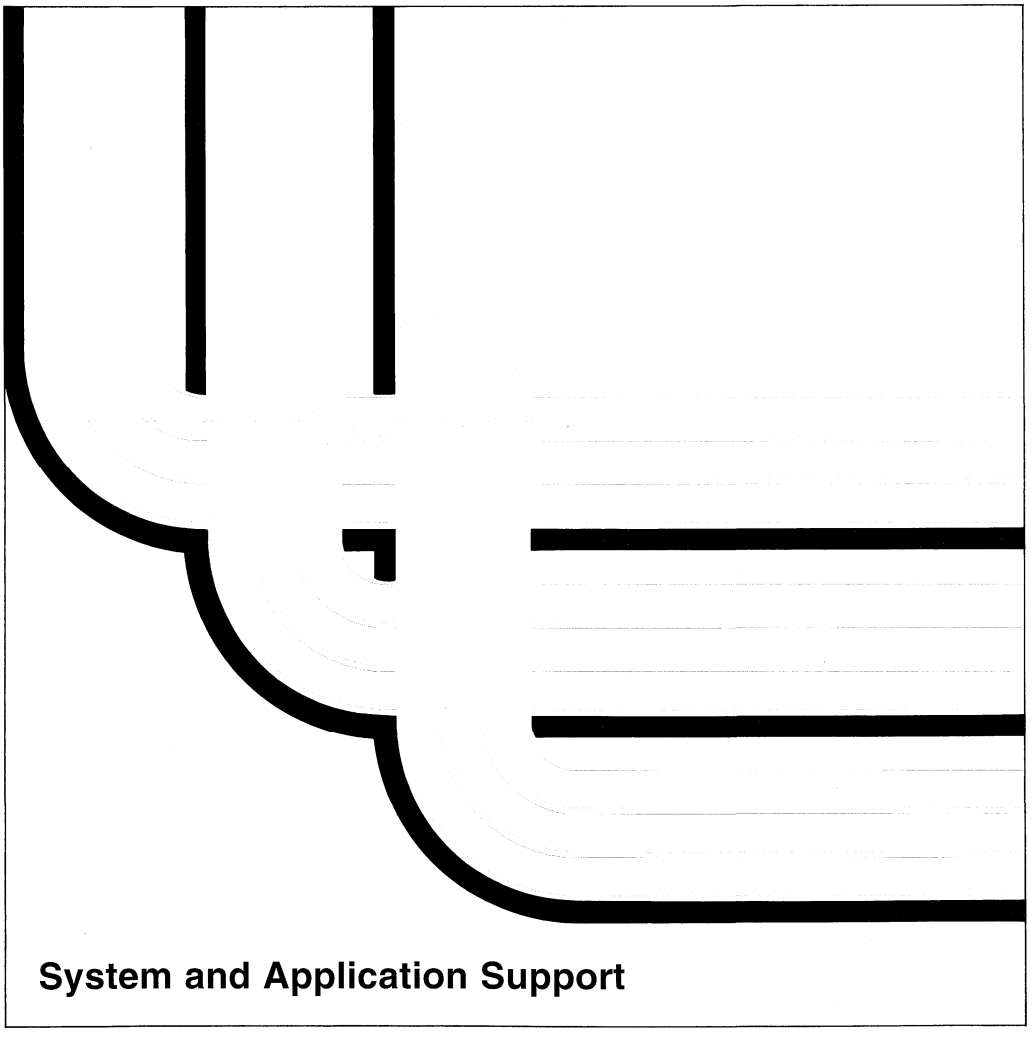
Application System/400

SC41-0030-02

Programming:
Control Language Reference
Common CL Information
ADDxxx through CPYxxx Commands



System and Application Support

Application System/400

SC41-0030-02

Programming:
Control Language Reference
Common CL Information
ADDxxx through CPYxxx Commands

Take Note!

Before using this information and the product it supports, be sure to read the general information under "Notices."

Third Edition (November 1993)

This edition applies to the licensed program IBM Operating System/400 (Program 5738-SS1), Version 2, Release 3 Modification 0, and to all subsequent releases and modifications until otherwise indicated in new editions. This major revision makes obsolete SC41-0030-01. Make sure you are using the proper edition for the level of the product.

Order publications through your IBM representative or the IBM branch serving your locality. Publications are not stocked at the address given below.

A Customer Satisfaction Feedback form for readers' comments is provided at the back of this publication. If the form has been removed, you can mail your comments to:

Attn Department 245
IBM Corporation
3605 Highway 52 N
Rochester, MN 55901-7899 USA

or you can fax your comments to:

United States and Canada: 800+937-3430
Other countries: (+1)+507+253-5192

Contents

Tables

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577, U.S.A.

This publication could contain technical inaccuracies or typographical errors.

This publication may refer to products that are announced but not currently available in your country. This publication may also refer to products that have not been announced in your country. IBM makes no commitment to make available any unannounced products referred to herein. The final decision to announce any product is based on IBM's business and technical judgment.

Changes or additions to the text are indicated by a vertical line (|) to the left of the change or addition.

Refer to the "Summary of Changes" for a summary of changes made to the licensed program IBM Operating System/400 (Program 5738-SS1), Version 2, Release 3 Modification 0 and how they are described in this publication.

This publication contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

This publication contains small programs that are furnished by IBM as simple examples to provide an illustration. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. All programs contained herein are provided to you "AS IS". THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED.

Trademarks and Service Marks

The following terms, denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation in the United States or other countries or both:

Advanced Function Printing	OfficeVision
Advanced Peer-to-Peer Networking	OfficeVision/400
AFP	Operating System/400
Application System/400	Operational Assistant
APPN	OS/2
AS/400	OS/400
BookManager	PrintManager
CICS	PROFS
COBOL/400	Proprinter
DB2	PS
DisplayWrite	PS/2
Distributed Relational Database Architecture	Q & A
DRDA	Quickwriter
Discover/Education	Quietwriter
ExecJet	RPG/400
GDDM	SAA
IBM	Selectric
ILE	SQL/400
Integrated Language Environment	Systems Application Architecture
Intelligent Printer Data Stream	SystemView
IPDS	System/370
MAPICS (only in U.S. publications)	VTAM
NetView	400

The following terms, denoted by a double asterisk (**) in this publication, are trademarks of other companies:

AT&T	AT&T
Century Schoolbook	American Type Foundry
COMPAQ	COMPAQ Computer Corp.
Dasher	Data General Corporation
Dataphone	American Telephone and Telegraph Company
DEC	Digital Equipment Corp.
DEFINITY	AT&T
DFM	Stratacom Inc.
Epson	Epson America Inc
Futura	Fundicion Tipografica Neufville, S. A.
Helvetica	Linotype AG or its subsidiaries
HP	Hewlett-Packard Company
LaserJet	Hewlett-Packard Company
Monotype Times New Roman	The Monotype Corporation plc.
NEC	NEC Corporation
Okidata	Okidata Corporation
Optima	Linotype AG or its subsidiaries
Palatino	Linotype AG or its subsidiaries
Panasonic	Matsushita Electric Industrial Co., Ltd.
RM/COBOL-85	Ryan McFarland Corporation
TeleVideo	TeleVideo Systems, Inc
Telex	Telex Computer Products Inc
Wyse	Wyse Corp.
WY30	Wyse Corp.
WY50	Wyse Corp.
WY60	Wyse Corp.
Xerox	Xerox Corporation
5ESS	AT&T

About This Manual

This manual describes the OS/400 control language (CL) commands. The information should be used as a reference to request functions from the OS/400 operating system. This manual includes *only* the command descriptions that are part of the base operating system. It does *not* include descriptions of commands that are part of licensed program products.

This manual does not describe all of the functions of the operating system. You may need to refer to other IBM manuals for more specific information about a particular topic. The *Publications Guide*, GC41-9678, provides information on all the manuals in the AS/400 library.

Who Should Use This Manual

This manual is intended for the programmer, data processing manager, and system operator.

To use this manual, you should have a general understanding of the IBM AS/400 operating system, and some background in programming.

Summary of Changes

This section lists new and deleted CL commands for this release, new object types, and new system values. It also describes significant parameter changes to CL commands that continue to be available from prior releases.

Many changes in this release are a result of the Integrated Language Environment (ILE) functions added to the OS/400 operating system to support a new family of high-level language (HLL) compilers. For more information on ILE features, see the *ILE* Concepts* publication.

New CL Commands

The following shows the new CL commands added to the OS/400 control language since Version 2 Release 2 Modification 0. This list shows new commands that are part of the OS/400 operating system.

Commands that are not part of the OS/400 operating system but are part of another licensed program are not included in this manual. (These commands are in the manual describing the related licensed program product. For example, the commands in the OfficeVision/400 licensed program product (5738WP1) are described in *Office Services Concepts and Programmer's Guide*.) To determine where these commands are documented, use the Master Command Matrix Table in the *Programming Reference Summary*.

The following commands are new in the OS/400 licensed program for Version 2 Release 3:

- Auditing commands
  - CHGDLOAUD (Change Document Library Object Audit)
  - CHGOBJAUD (Change Object Auditing)
  - CHGUSRAUD (Change User Auditing)
  - DSPDLOAUD (Display Document Library Object Audit)
- Binding modules and service programs commands
  - ADDBNDDIRE (Add Binding Directory Entry)
  - CHGMOD (Change Module)
  - CHGSRVPGM (Change Service Program)
  - CRTBNDDIR (Create Binding Directory)
  - CRTPGM (Create Program)
  - CRTSRVPGM (Create Service Program)
  - DSPBNDDIR (Display Binding Directory)
  - DSPMOD (Display Module)
  - DSPSRVPGM (Display Service Program)
  - DLTBNDDIR (Delete Binding Directory)
  - DLTMOD (Delete Module)
  - DLTSRVPGM (Delete Service Program)
  - RMVBNDDIRE (Remove Binding Directory Entry)
  - WRKBNDDIR (Work with Binding Directories)
  - WRKBNDDIRE (Work with Binding Directory Entries)
  - WRKMOD (Work with Modules)
  - WRKSRVPGM (Work with Service Programs)
- Communications configuration commands
  - CHGLINDDI (Change Line Description (DDI Network))
  - CHGLINFAX (Change Line Description (Fax))
  - CHGLINFR (Change Line Description (Frame Relay Network))
  - CHGNWIFR (Change Network Interface (Frame Relay Network))
  - CRTLINDDI (Create Line Description (DDI Network))
  - CRTLINFAX (Create Line Description (Fax))
  - CRTLINFR (Create Line Description (Frame Relay Network))
  - CRTNWIFR (Create Network Interface (Frame Relay Network))
- License information commands
  - CHGLICINF (Change License Information)
  - WRKLICINF (Work with License Information)
- Performance monitor commands
  - ENDPFRCOL (End Performance Collection)
  - STRPFRCOL (Start Performance Collection)
- SNA node list commands
  - ADDNODLE (Add Node List Entry)
  - CRTNODL (Create Node List)
  - DLTNODL (Delete Node List)
  - RMVNODLE (Remove Node List Entry)
  - WRKNODL (Work with Node Lists)
  - WRKNODLE (Work with Node List Entries)
- System/36 environment commands
  - CHGS36A (Change System/36 Attributes)
  - RTVS36A (Retrieve System/36 Attributes)
- Miscellaneous commands:
  - DSPMODSRC (Display Module Source)
  - INZDSTQ (Initialize Distribution Queue)
  - RCLACTGRP (Reclaim Activation Group)
  - RQSORDAST (Request Order Assistance)

Deleted Command

The following command has been removed from the OS/400 licensed program for Version 2 Release 3.

- CRTAPAR (Create Authorized Program Analysis Report)

The Save APAR Data (SAVAPARDTA) command is available for your use in lieu of the CRTAPAR command.

New Object Types

The following object types are new in the OS/400 licensed program for Version 2 Release 3:

- *BNDDIR (binding directory)
- *MODULE (module)
- *NODL (node list)
- *SRVPGM (service program)

These new object types are described in the new binding modules and service programs commands and SNA node lists commands listed in "New CL Commands."

New System Values

The following system values are new in the OS/400 licensed program for Version 2 Release 3:

- QALWUSRDMN (Allow user domain objects in libraries)
- QAUDENDACN (Auditing end action)
- QAUDCTL (Auditing control)
- QAUDFRCLVL (Force auditing data)
- QCRTOBJAUD (Create object auditing)
- QJOBMSGQFL (Job message queue full option)
- QJOBMSGQMX (Maximum size of job message queue)
- QSRTSEQ (Sort sequence)

More information about system values is in the *Work Management Guide*.

Significant Areas of Parameter Changes and Additions

Following are the significant areas of parameter and value changes made to existing commands.

Note: The parameter and value additions in each of the categories below do not necessarily affect all of the commands in the category. Some parameters are added to many of the commands in the category, while others are added only to a few. Details describing the specific parameters and values added to each command can be found in the description for each command in Part 3, "OS/400 CL Commands."

- Communications commands:
  - On *communications line* commands (such as xxxLIND, xxxLINETH, xxxLINSDLC, and xxxLINTRN) several parameters are added including CLOCK, NWI, NWIDLCI, and NWITYPE. With the CLOCK parameter, you can improve high speed transmission on a synchronous data link control line. With the NWI parameters, you can specify frame relay networking support.
  - On *controller* commands (such as xxxCTLHOST) the AUTOCRTDEV parameter is added. With this parameter, you can create automatically the devices that you need for your Advanced Peer-to-Peer Networking controller.
  - On *device* commands (such as xxxDEVD, xxxDEVDSP, xxxDEVHOST, xxxDEVPRT, xxxDEVTAP) several parameters are added.
    - You can use the ENDSSNHOST parameter to specify how your host device will end remote sessions.
    - You can use the ASSIGN and UNLOAD parameters to assign tape drives when you vary on the system and to specify whether to unload the tape drive when you vary off the system.
    - You can use the TRANSFORM, MFRTYPMDL, PPRSRC1, PPRSRC2, ENVELOPE, and ASCII899 parameters to have your AS/400 system support ASCII printers without using an emulator.
    - You can use the SEPPGM parameter call a user exit program when you are printing job and file separators.
    - You can use the LCLLOCNAME, RMTLOCNAME, and RMTNETID parameters to support LU2 display stations and LU1 printers through a Systems Network Architecture (SNA) subarea network.
  - On *network interface* commands (such as xxxNWIISDN) the SPID parameter and the *ETSI, NISDN, *DCLI, and *PCLENTRY values are added to conform to the National ISDN standard.
  - Several new values are added to the *communications* commands:
    - The *CTLSSN, *DEVINIT, and *APPINIT values are added to provide additional LU1 and LU2 SNA support.
    - The *DDI, *FAX, and *FR values are added so you can work with the lines created with the new distributed data interface, facsimile, and frame relay line description commands.
    - More values for line speeds and link speeds are added.
- Create commands (such as CRTxxx): The *ALL value on the AUT parameter is clarified. The user can change ownership of the object.
- File, job, and user profile commands (such as xxxF, xxxJOB, xxxUSRPRF): The CNTRYID, LANGID, and SRTSEQ parameters are added so you can sequence your data easily.
- Network attribute and mode description commands (such as xxxNETA and xxxMODD): The DTACPR, DTACPRINM, INDTACPR and OUTDTACPR parameters are added so you can compress data at the level you specify to meet your speed and storage needs.
- Override commands (such as xxxOVR and OVRxxx): The LVL, OVRSCOPE, and OPNSCOPE parameters are added so you can specify the extent of influence of overrides.
- Performance monitor commands (such as xxxPFRMON and xxxPFRCOL): The EXITPGM parameter is added

so you can call an exit program when the performance monitor ends.

- Save and restore commands:
  - On *save* commands (such as SAVxxx) the *SYNCLIB value is added to the SAVACT parameter so you can minimize the amount of time the system is unavailable when you are running daily backups.

    The MAIL parameter is removed from the SAVSECDTA command.
- On *restore* commands (such as RSTxxx) the OUTFILE, OUTMBR, and INFTYPE parameters are added so you can get information about a restore operation at the level of detail you specify.

Part 1. Basic AS/400 Control Language Information

Chapter 1. Overview of OS/400 Object Types and CL Commands

This chapter briefly reviews the concept of an object on the Application System/400* (AS/400*) system and how you can manipulate objects using control language (CL) commands. For a more detailed introduction to the system, see the *System Introduction* and the *System Concepts* publications.

OS/400 Object Types

Operating System/400* (OS/400) objects provide the means through which all data processing information is stored and processed by the AS/400 system. An *OS/400 object* is a named unit that exists (occupies space) in storage and can have operations performed on it by the operating system. Each object has a set of descriptive attributes that are defined when the object is created. The name of the object to be used by the system to perform a specific function must be specified in the CL command that performs that function.

Several types of OS/400 objects are created and used in the control language. They are identified in two tables in "OBJTYPE Parameter" in Appendix A. Additional information about each object type is in the *Programming Reference Summary*; it gives the object type, the system-recognized identifier for the object type, and a brief description of its purpose in the OS/400 licensed program.

All objects have a set of attributes that describe the object and give the specific values assigned for those attributes.

Generally, each object is independent of all other objects. However, some objects must be created before other objects can be created; for example, a logical file cannot be created if the physical file it must be based on does not exist. Each object must be created before other operations that use the object are performed.

Descriptions of the create (CRT) commands give more information about the object types that they create.

Most objects are grouped in special objects called libraries. More information about the use of libraries is in the *CL Programmer's Guide*. A list of IBM-supplied libraries is provided in the *Programming Reference Summary*.

Commands Operating on OS/400 Objects

Each of the OS/400 object types has a set of commands that operates on that object type. Most OS/400 object types have commands that perform the following actions:

- Create (CRT): Creates the object and specifies its attributes.
- Delete (DLT): Deletes the object from the system.
- Change (CHG): Changes the attributes and/or contents of the object.
- Display (DSP): Displays the contents of the object. Display commands cannot be used to operate on objects.
- Work with (WRK): Works with the attributes and/or contents of the object. Unlike display commands, work commands allow users to operate on objects and modify applications.

How Commands Are Named: Command names consist of a combination of the verb and the object being acted on: (command = verb + object acted on). For example, you can create, delete, or display a class; so the verb abbreviations CRT, DLT, and DSP are joined to the abbreviation for class, CLS. The result is the three commands that can operate on a class: CRTCLS, DLTCLS, and DSPCLS.

The IBM-supplied commands are named in a consistent manner. The primary convention (as just shown) is that three letters from each word in the descriptive command name are used to form the abbreviated command name that is recognized by the system. The secondary convention is that single letters from the *ending* word or words are used for the end of the command name, such as the three single letters DLO on the DLTDLO (Delete Document Library Object) command. There are, however, a few exceptions where single letters are used in the middle of some command names (usually between commonly used three-character verbs and objects), such as the letters CL in the CRTCLPGM (Create CL Program) command.

For a complete list of all the abbreviations used in command (and keyword) names, see the *Programming Reference Summary*.

Commands Operating on Multiple Objects

In addition to the commands that operate on single object types, there are commands that operate on several object types; for example:

- Display object description (DSPOBJD): Displays the common attributes of an object.
- Move object (MOVOBJ): Moves an object from one library to another.
- Rename object (RNMOBJ): Specifies the new name of an object.
- Save object (SAVOBJ): Saves an object and its contents on diskette, tape, or in a save file.
- Restore object (RSTOBJ): Restores a saved version of the object from diskette, tape, or from a save file.

Table 1 shows the commands that perform an action on many of the object types. Some of the commands, such as the Move Object (MOVOBJ) command, operate on only one

object at a time, but that object can be any one of several OS/400 object types. For example, the MOVOBJ command can move a file or a job description.

Table 1. Commands Operating on Multiple Object Types

Item	Identifier	Commands
Object	OBJ	ALC, DLC, SAV, RST, CHK, MOV, RNM, DMP, CRTDUP
Object Authority	OBJAUT	DSP, GRT, RVK, EDT
Object Description	OBJD	DSP
Object Lock	OBJLCK	WRK
Object Owner	OBJOWN	CHG

Also, you can refer to Table 81 on page P4-13 in "OBJTYPE Parameter" in Appendix A to see how these multiple-object commands affect specific object types.

Commands Operating on Multiple Object Types at Once

Some commands are more powerful, such as the Display Object Description (DSPOBJD) command, because they can operate on several objects of different types at the same time. By specifying multiple objects in a single DSPOBJD command, you can display the object descriptions of a group of objects.

Finding CL Command Descriptions

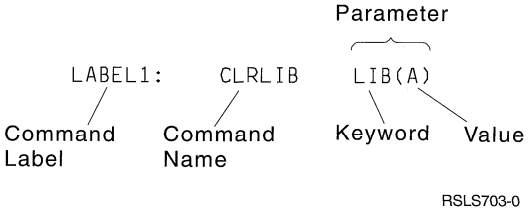
For a list of all the CL commands and where they can be found in the AS/400 library, see the *Programming Reference Summary*. It contains the master command matrix table that has the names of *all* the commands for the OS/400 licensed program and all the commands supported in the other AS/400 licensed programs. The OS/400 commands are described in this manual (*CL Reference*) and the non-OS/400 commands are described in the manuals related to the other AS/400 licensed programs. The *Programming Reference Summary* is your master index that tells you which manual each command is described in.

Chapter 2. Control Language Syntax

This chapter describes the control language syntax that must be used to code and enter AS/400 control language (CL) commands. Each CL command is processed by Operating System/400 (OS/400) to perform the specified command function upon objects named in the command.

Parts of a Command

A CL command has the following parts: command label (optional), command name (mnemonic), and one or more parameters. A parameter includes a keyword and a value.



Command Label

Command labels identify particular commands for branching purposes in a CL program. Labels can also be used to identify statements in CL programs that are being debugged: they can identify statements used either as breakpoints or as starting and ending statements for tracing purposes.

A command label is typed just before the command name. The label can contain as many as 10 characters and follows the standard rule for specifying simple names (*SNAME). Refer to "Rules for Specifying Names" later in this chapter. The label must be immediately followed by a colon, and blanks (though not required) can occur between the colon and the command name. START: and TESTLOOP: are examples of command labels.

Command labels are not required, but a label can be placed on any command. If a label is placed on a command that cannot be run (for example, the Declare CL Variable (DCL) command) and the program branches to that label, the next command following the label is run. If the command following the label cannot be run, the program moves to the next command that can be run. Similarly, only one label can be specified on a line; if no command is located on that line, the program jumps to the next command that can be run.

To specify multiple labels, each additional label must be on a separate line preceding the command as shown:

```
LABEL1:
LABEL2:  CMDX
```

No continuation character (+ or –) is allowed on the preceding label lines.

Command Name

The command name identifies the function to be performed by the program that is called when the command is run. The command name is an abbreviation of the command description; for example, the name MOVOBJ identifies the CL command (Move Object) that moves an object from one library to another. Like other objects, a command name can be optionally qualified by a library name. Simple and qualified object names are described later in this chapter.

The IBM-supplied commands are all named in a consistent manner. Generally, three letters from each word in the descriptive command name are used to form the abbreviated command name that is recognized by the system.

Command Parameters

Most CL commands have one or more parameters that specify the objects and values used to run the commands. When a command is entered, the user supplies the object names, the parameter name, and the parameter values used by the command. The number of parameters specified depends upon the command. Some commands (like DO and ENDBCHJOB) have no parameters, and others have one or more.

A *parameter* identifies an individual value or group of values used by the command. The specification of a group of values on one parameter is described under "Lists of Values" later in this chapter.

In this reference manual, the word *parameter* usually refers to the combination of the parameter keyword and its value. For example, the Move Object (MOVOBJ) command has a parameter called OBJ that requires an object name to be specified. OBJ is the parameter keyword, and the name of the object is the value entered for the OBJ parameter.

A command can have parameters that must be coded (required parameters) and parameters that do not have to be coded (optional parameters). Optional parameters are usually assigned a system-defined default value if another value is not specified for the parameter when the command is entered.

A command can also have *key parameters* which are the only parameters shown on the display when a user prompts for the command. After values are entered for the key parameters, the remaining parameters are shown with actual values instead of the default values (such as *SAME or *PRV).

Note: Key parameters include all parameters preceding the key parameter limit, which is designated on the syntax diagram by the letter ***K***.

Parameters in CL can be specified either in keyword form or positional form or in a combination of the two. Descriptions of these forms follow.

Parameters in Keyword Form: A parameter in *keyword form* consists of a keyword followed immediately by a value (or a list of values separated by blank spaces) enclosed in parentheses. No blanks may be placed between the keyword and the left parenthesis preceding the parameter value. Blanks may be placed between the parentheses and the parameter value. For example, LIB(MYLIB) is a keyword parameter specifying that MYLIB is the name of the library that is used in some way, depending upon the command in which this LIB parameter is used.

When command parameters are all specified in keyword form, they can be placed in any order. For example, the following two commands are the same:

```
CRTLIB  LIB(MYLIB)  TYPE(*TEST)
CRTLIB  TYPE(*TEST)  LIB(MYLIB)
```

Parameters in Positional Form: A parameter in *positional form* does not have its keyword coded; it contains only the value (or values, if it is a list) whose function is determined by its position in the parameter set for that command. The parameter values are separated from each other and from the command name by one or more blank spaces. Because there is only one positional sequence in which parameters can be coded, the positional form of the previous CRTLIB example is:

```
CRTLIB  MYLIB  *TEST
```

Most commands with more than one parameter can have their parameters coded in a specific positional order. The correct order is shown in the syntax diagram. However, in the few cases where dependent parameters occur in the syntax diagram, or when one parameter or another (but not both) must be used and the positional order is not readily apparent, the correct order can be easily determined from the text, because the parameter descriptions in this book are described in positional order.

If you do not want to enter a value for one of the parameters, the predefined value *N (null) can be entered in that parameter's position. The system recognizes *N as an omitted parameter, and either assigns a default value or leaves it null. In the previous CRTLIB command example, if you coded *N instead of *TEST for the TYPE parameter, the default value *PROD is used when the command is run, and a production library named MYLIB is created with no public authority. The description of the CRTLIB command contains the explanation for each parameter.

Note: Parameters may not be coded in positional form beyond the positional coding limit, which is designated in the syntax diagrams with the letter ***P***.

If you attempt to code parameters in positional form beyond that point, the system returns an error message. When all parameters of a command can be coded in positional form or when parameters *do not* exist on a command, no positional limit symbol appears in the syntax diagram. When a command has parameters, but none of its parameters can be specified in positional form, the letter ***P*** appears immediately to the right of the command name in the syntax diagram.

If the parameters that can be coded in positional form are the same as the key parameters, both the ***P*** and the ***K*** will be shown on the diagram as follows: ***(P,K)***.

Entering Parameters in Both Forms: A command may also have its parameters coded in a combination of keyword and positional forms. The following examples show three ways to code the Declare CL Variable (DCL) command.

Keyword form:

```
DCL   VAR(&QTY)  TYPE(*DEC)  LEN(5)  VALUE(0)
```

Positional form:

```
DCL   &QTY  *DEC  5  0
```

Positional and keyword forms together:

```
DCL   &QTY  *DEC  VALUE(0)
```

In the last example, because the optional LEN parameter was not coded, the VALUE parameter *must* be coded in keyword form.

Note: Once positional coding is discontinued, it cannot be started again.

Command Syntax

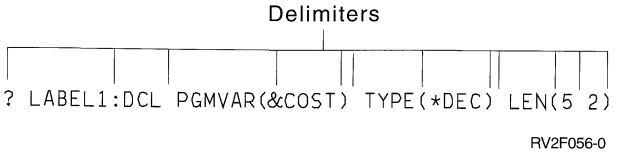
Commands have the following general syntax. The brackets indicate that the item within them is optional; however, the parameter set may or may not be optional, depending upon the requirements of the command.

```
[//] [?] [label-name:][library-name/]command-name
[parameter-set]
```

Note: The // is valid only for a few batch job control commands, such as the DATA command. The // identifies those types of commands sent to the spooling reader that reads the batch job input stream. The remaining parts of the command are described in the sections that follow.

Command Delimiters

Command delimiters are special characters or spaces that identify the beginning or end of a group of characters in a command. Delimiters are used to separate a character string into the individual parts that form a command: command label, command name, parameter keywords, and parameter values. Parameter values can be constants, variable names, lists, or expressions.



The following delimiters are used in the OS/400 control language:

- The colon (:) ends the command label, and it can be used to separate the command label from the command name.
- Blank spaces separate the command name from its parameters and separate parameters from each other. They also separate values in a list. Multiple blanks are treated as a single blank except in a quoted character string or comment enclosed in apostrophes. A blank *cannot* separate a keyword and the left parenthesis for the value.
- Parentheses ( ) are used to separate parameter values from their keywords, to group lists of values, and to group lists within lists.
- Slashes (/) connect the parts of a qualified name. For a qualified object name, the two parts are the library qualifier and the object name (LIBX/OBJA). Qualified object names are described in "Identifying OS/400 Objects" later in this chapter.
- Either a period or a comma can be used as a decimal point in a decimal value (3.14 or 3,14). Only one decimal point is allowed in a value.
- Apostrophes specify the beginning and end of a quoted character string, which is a combination of any of the 256 extended binary-coded decimal interchange code (EBCDIC) characters that are used as a constant. For example, 'YOU CAN USE $99@123.45 ()*></ and lowercase letters' is a valid quoted string. Because an apostrophe inside a quoted string is paired with the opening apostrophe (delimiter) and is interpreted as the ending delimiter, an apostrophe inside a quoted string must be specified as two apostrophes. A pair of adjacent apostrophes used this way is counted as a single character.
- A special character is used to separate a date into three parts: month, day, and year (two parts for Julian dates: year and day). The special characters that may be used as date separators are the slash (/), the hyphen (-), the period (.), and the comma (,). The special character used to code as separators in a command date must be the same as the special character specified as the date separator for the job. The *Work Management Guide* contains more information on specifying job and system values.
- The characters /* and */ indicate the beginning and end of a comment. To begin a comment, the characters /* must begin in the first position of the command, be preceded by a blank, or be followed by either a blank or an asterisk.
- A question mark (?) preceding the command name indicates that the command is prompted. If the command is specified with a label, the question mark may either precede the label, or it may follow the label and precede the command name.

  Within a CL program, when a question mark precedes a command name, a prompt display is presented. You can enter parameter values not specified on the command in the program.

  Prompting characters may be put into a command in two forms. A single question mark (?) may be coded before the command name (either before or after the command label in a CL program) to cause the entire command to be prompted. Selective prompt characters (?? or ?*) may be coded before any parameter keyword to cause that parameter to be prompted when the command is run.

  If a question mark is entered before the command name on the command entry display, the effect is the same as pressing the F4 (Prompt) key after the command is entered.

  Within a CL program, when a question mark precedes the command name, a prompt display is presented. This display is of the same format as that presented when pressing the F4 key from the command entry display. Parameters of the command for which the program has coded values are shown for informational purposes, but the user cannot change the values supplied by the program. Parameters for which no value was coded are shown as input fields so you can enter values to be used in processing the command.

  Selective prompting allows you to identify specific command parameters to be prompted. To call selective prompting, the characters ??, ?*, or ?– are coded immediately preceding the keyword name of the parameter(s) to be prompted. Additional information on selective prompting is in the *CL Programmer's Guide*. Parameters of the command that are preceded by the characters ?* are shown, but you cannot change the values that are supplied by the program. Parameters preceded by the characters ?? are shown as input fields containing the values coded in the program or command defaults so you can enter or change the values used in processing the command. Parameters preceded by the characters ?- are omitted from the display. All selectively prompted parameters must be coded in keyword or keyword-with-value form. Several parameters may be selectively prompted within one command. When selective prompting is called, only keywords that are immediately preceded by the selective prompt characters are prompted. All other parameters are processed using the values as coded on the command or, if not coded, using command defaults.

  Either form of prompting, but not both, is allowed on a single command in a CL program. If the character ? precedes the command name and selective prompt

characters (except ?–) precede any keyword, an error message is returned and the program is not created.

More information on prompting is described in the *CL Programmer's Guide* and the *New User's Guide*.

Command Continuation

Commands can be entered in free format. This means that a command does not have to begin in a specific location on a coding sheet or on the display. A command can be contained entirely in one record, or it can be continued on several lines or in several records. Whether continued or not, the total command length cannot exceed 3000 characters. Either of two special characters, the plus sign (+) or the minus sign (–), is entered as the last nonblank character on the line to indicate that a command is continued. Blanks immediately *preceding* a + or – sign are always included; blanks immediately following a + or – in the *same record* are ignored. Blanks in the *next record* that precede the first nonblank character in the record are ignored when + is specified but are included when – is specified.

The + is generally useful between parameters or values. At least one blank must precede the sign when it is used between separate parameters or values. The difference between the plus and minus sign usage is particularly important when continuation occurs inside a quoted character string. The following example shows the difference:

```
CRTLIB LIB(XYZ) TEXT('This is CONT+
bbbINUED')

CRTLIB LIB(XYZ) TEXT('This is CONT-
bbbINUED')

For + : CRTLIB LIB(XYZ) TEXT('This is CONTINUED')

For - : CRTLIB LIB(XYZ) TEXT('This is CONTbbbINUED')
```

Notes:

1. The minus sign causes the leading blanks on the next line to be entered.
2. Use continuation characters + and - in CL programs only. An error occurs if + or - is used on a command entry display.
3. In this manual, + and - are used for multiple-command examples, but not for single-command examples.

Entering Comments

Comments can be inserted either inside or outside a command's character string wherever a blank is permitted. However, because a continuation character must be the last nonblank character of a line (or record), comments *may not* follow a continuation character on the same line.

For readability, it is recommended that each comment be specified on a separate line preceding or following the command it describes, as shown here:

```
MOVOBJ  OBJA  TOLIB(LIBY)
        /* Object OBJA is moved to library LIBY. */
DLTLIB  LIBX
        /* Library LIBX is deleted. */
```

Comments can include any of the 256 EBCDIC characters. However, the character combination */ should not appear within a comment because these characters end the comment. To begin a comment, the characters /* must be placed in the first position of the command, be preceded by a blank, or be followed by either a blank or an asterisk.

Control Language Character Set

The control language uses the extended binary-coded decimal interchange code (EBCDIC) character set. For convenience in describing the relationship between characters used in the control language and those in the EBCDIC character set, the following control language categories contain the EBCDIC characters shown:

Category	Characters Included
Alphabetic[1]	26 letters (A through Z), $, #, and @
Numeric	10 digits (0-9)
Alphanumeric[2]	A through Z, 0 through 9, $, #, @, period (.), and underscore (_)
Special Characters	All other EBCDIC characters (for those having special uses in CL, see "Summary of Special Character Usage")

1 Lowercase letters (a through z) are accepted, but they are translated into the corresponding uppercase letters by the system except when they are included within a quoted character string or a comment. In the Katakana EBCDIC character set, the character positions corresponding to a-z in the US character set contain Katakana characters that can be used as data in quoted strings or comments. If those characters are used outside quoted strings or comments, they are translated to character positions corresponding to A through Z in the US.

2 Lowercase letters (a through z) are accepted, but they are translated into the corresponding uppercase letters by the system except when they are included within a quoted character string or a comment. The underscore (_) is an alphanumeric connector that can be used in AS/400 CL to connect words or alphanumeric characters to form a name (for example, PAYLIB_01). This use of the underscore might not be valid in other high-level languages.

The first three categories contain the characters that are allowed in quoted and unquoted character strings, in comments, and in CL names, such as in names of commands, labels, keywords, variables, and OS/400 objects. Special characters in the last category can only be used in quoted character strings and comments; they cannot be used in unquoted strings. However, some have special syntactical uses when coded in the proper place in CL commands. These uses are given in the chart in "Summary of Special Character Usage" under "Special Characters and Predefined Values" that follows in this chapter.

Special Characters and Predefined Values

This section summarizes in chart form all of the AS/400 special characters and their uses in the control language. A description of predefined values and how they are used is also included.

Summary of Special Character Usage

The following special EBCDIC characters are used by the CL in various ways. They are most frequently used as delimiters (which were covered in "Command Delimiters" under "Command Syntax" in this chapter) and as symbolic operators in expressions (see Chapter 3, "Expressions in CL Commands"). Special characters can only be used in these special ways or inside quoted character strings or comments. The special characters have the following assigned meanings when coded in control language commands:

Delimiters

Name	Symbol	Meanings
Apostrophe	' '	Single apostrophe delimiters indicate the beginning and end of a quoted character string (a constant).
Begin and end comment	/* */	Indicates the beginning and end of a comment.
Blank	ƀ[1]	Basic delimiter for separating parts of a command (label, command name, and its parameters), and for separating values inside lists.
Colon	:	Ending delimiter for command labels. Separates parts of time values.
Comma	,	In many countries, used as decimal point in numeric values. Separates parts of date values.[2]
Left and right parentheses	( )	Grouping delimiter for lists and parameter values, and for evaluating the order of expressions.
Period	.	Decimal point. Used to separate the name and extension of a document and folder name and to separate the parts of date values.[2]
Quote	" "	Start of a quoted object name.
Slash	/	Connects parts of qualified names.
Slashes	//	Identifying characters used in positions 1 and 2 of BCHJOB, ENDJOB, and DATA commands in the job stream. Also, used as a default delimiter on inline data files.

1 In this manual, ƀ is used when necessary to represent a blank space.
2 Valid only when the job date separator value specifies the same character.

Symbolic Operators

Name	Symbol	Meanings
And	&	Symbolic logical operator for AND.
Asterisk	*	Multiplication operator. Indicates a generic name when it is the last character in the name. Indicates OS/400 reserved values (predefined parameter values and expression operators) when it is the first character in a string.
Concatenation	|>, |<, and ||[3]	Character string operator (indicates both values are to be joined). See Chapter 3, "Expressions in CL Commands" for more information on the differences between these concatenation operators.
Equal	=	Symbolic *equal* relational operator.
Greater than	>	Symbolic *greater than* relational operator.
Less than	<	Symbolic *less than* relational operator.
Minus (hyphen)	-	Subtraction operator, command continuation operator, and negative signed value indicator. Separates parts of date values.[1]
Not	¬[2]	Symbolic NOT relational operator.
Or	|[3]	Symbolic logical operator for OR.
Plus	+	Addition operator, command continuation character, and positive signed value indicator.

Name	Symbol	Meanings
Slash	/	Division operator. Separates parts of date values.[1] Used as the separator between parts of a qualified name.

1 Valid only when the job date separator value specifies the same character.
2 In some character sets, including the multinational character set, the character ∧ replaces the ¬ character. Either ∧ or *NOT can be used as the logical NOT operator in those character sets.
3 In some character sets, including the multinational character set, the character ! replaces the | character. Either ! or *OR can be used as the logical OR operator, and either !! or *CAT can be used as the concatenation operator in those character sets.

Note: The symbolic operators can also be used in combinations as listed in the chart under "Operators in Expressions" in Chapter 3, "Expressions in CL Commands."

Other Uses

Name	Symbol	Meanings
Ampersand	&	Identifies a CL variable name when it is the first character in the string.
Percent	%	Identifies a built-in system function when it is the first character in the string.
Question mark	?	Specifies a prompt request when it precedes a command name or keyword name.

Predefined Values

Predefined values are IBM-defined fixed values that have predefined uses in the CL and are considered to be reserved in OS/400. Predefined values have an asterisk (*) as the first character in the value followed by a word or abbreviation, such as *ALL or *PGM. The purpose of the * in predefined values is to prevent possible conflicts with user-specified values, such as object names. Each predefined value has a specific use in one or more command parameters, and each is described in detail in the command description.

Some predefined values are used as operators in expressions, such as *EQ and *AND. The predefined value *N is used to specify a null value and can be used to represent any optional parameter. A *null value* (*N) indicates a parameter position for which no value is being specified; it allows other parameters that follow it to be entered in positional form. To specify the characters *N as a character value (not as a null), the string must be enclosed in apostrophes ('*N') to be passed. Also, when the value *N appears in a CL program variable at the time it is run, it is always treated as a null value.

Rules for Specifying Names

The type of name you specify in the OS/400 control language determines the characters you can use to specify a name. For certain types of names, there are restrictions on the use of certain characters to represent the name. Those types of names are *NAME, *SNAME, *CNAME, and *GENERIC.

Note: For a description of how to specify these names when you use command definitions to create commands, see the PARM (parameter) and ELEM (element) statements in Chapter 5, "Command Definition Statements."

The characters allowed for these names and the rules you use to specify them are shown in the table and descriptions that follow.

*Table 2. Allowable Characters for *NAME, *SNAME, and *CNAME.*

Type of Name	First Character	Other Characters	Min. Length	Max. Length
*NAME[1]	A-Z, $, #, @	A-Z, 0-9, $, #, @, _, .	1	256
*SNAME[1]	A-Z, $, #, @	A-Z, 0-9, $, #, @, _	1	256
*CNAME[1]	A-Z, $, #, @	A-Z, 0-9, $, #, @	1	256
Quoted name[2]	"[3]	Any except blank, *, ?, ', ", X'00'-X'3F', and X'FF'	3	256

Notes:
1 The system converts lowercase letters to uppercase.
2 Double quotes can only be used for basic names (*NAME).
3 Both the first and last characters must be a double quote (").

Reprinted with the permission of *News 3X/400.*

***NAME (Basic Name):** Every basic name can begin with the characters A-Z, $, #, or @ and can be followed by up to nine characters. The remaining characters can include the same characters as the first but can also include numbers 0-9, underscores (_), and periods (.). Lower case letters (a-z) are changed to upper case letters (A-Z) by the system. Basic names used in IBM-supplied commands can be no longer than 10 characters. However, in your own commands, you can define parameters of type *NAME (specified on the TYPE parameter of the PARM or ELEM statements) with up to 256 characters.

Examples of basic names are shown below:

```
A987@.442#    ONE_NAME    LIBRARY_0690    $LIBX
```

Names can be entered in quoted or unquoted form. If you use the quoted form, the following rules and considerations also apply:

***NAME (Basic Name in Quoted Form):** Every quoted name must begin and end with a quotation mark ("). The middle characters of a quoted name can contain any character except b, *, ?, ', ", hex 00 through 3F, or hex FF, and is delimited by a slash. Quoted names allow you to use graphic characters in the name. The quoted form of basic names used in IBM-supplied commands can be no longer than 8 characters (not including the double quotes). In your own commands, you can define parameters of type *NAME in quoted form with up to 254 characters (not including the double quotes).

Note: Only basic names can be used in quoted form.

Examples of quoted names are shown below:

```
"A"        "AA%abc"        "ABC%%abc"
```

When you use quoted names, you should be aware of certain restrictions:

- Code points in a name might not be addressable from all keyboards.
- Characters in a quoted name might not be valid in a high-level language.
- The System/38 environment supports only simple (*SNAMES) names. If other characters are used, the objects are accessed as System/38 environment objects.
- Names that are longer than eight characters cannot be accessed by the System/36 environment unless control language overrides are used.
- A Structured Query Language (SQL) name that contains a period must be specified in an SQL statement in quotation marks.

If a name enclosed in quotation marks is a valid unquoted basic name, the quotation marks are removed. Thus, "ABC" is equivalent to ABC. Because the quotation marks are removed, they are not included in the length of the name. "ABCDEFGHIJ" is, therefore, a valid name on IBM* commands even though it is longer than 10 characters.

***SNAME (Simple Name):** Simple names are the same as *unquoted* basic names but with one exception: periods (.) cannot be used. Simple names are used for CL variables, labels, and keywords to simplify the syntax of the control language.

Some examples of simple names are as follows:

```
NEWCMD    LIB_2
```

***CNAME (Communications Name):** Communications names are the same as *unquoted* basic names but with the following exceptions:

1. Periods (.) and underscores (_) cannot be used.
2. For IBM commands, *CNAME is limited to 8 characters.

An example of a communications name is shown below:

```
APPN3@@
```

Note: Because restricted character sets are sometimes used by other IBM systems, use caution when choosing names that use the special characters #, $, and @. These special characters might not be on the remote system's keyboard. The names that may be exchanged with the remote systems include the following:

- Network IDS
- Location names
- Mode names
- Class-of-service names
- Control point names

***GENERIC (Generic Name):** A generic name is one that contains at least one initial character that is common to a group of objects, followed by an asterisk. (The asterisk identifies the series of common characters as a generic name; otherwise, the system interprets the series of characters as the name of a specific object).

When you specify a generic name, the system performs the required function on all objects whose names begin with the specified series of characters. You must have the authority required to perform that function on every object the generic name identifies; if you do not, the function is not performed for that object and a diagnostic message is issued for each instance in which the attempted generic function failed. A completion message is issued for each object the generic function operates on successfully. You must view the online low-level messages to see the completion messages. For example, from the DSPJOBLOG display, press the F10 key to view the low-level messages. Once the entire generic function is completed, a completion message is issued that states that all objects were operated on successfully, or an escape message is issued that states that one or more objects could not be successfully operated on. If an override is in effect for a specific device file, the single object name specified on the override is used to perform the operation, rather than the generic name.

You may not be able to use a generic name for delete, move, or rename commands if the library containing the objects is already locked. A search for generic object names requires a more restrictive lock on the library containing the objects than a search for full object names. The more restrictive lock is necessary to prevent another user from creating an object with the same name as the generic search string in the library while the delete, move, or rename command is running. You can circumvent this problem by using the full name of the objects instead of a generic name. Or you can end the job or subsystem that has a lock on the library.

Note: Use the WRKOBJLCK (Work with Object Locks) command to determine which jobs or subsystems have a lock on the library.

For some commands, a library qualifier can be specified with the generic name to limit the scope of the operation. For

example, a Change Print File (CHGPRTF) command with FILE(LIB1/PRT*) performs the desired operation on printer files beginning with PRT in library LIB1 only; printer files in other libraries are not affected.

The limitations associated with the various library qualifiers are as follows:

- *library-name:* The operation is performed on generic object names only in the specified library.
- *LIBL: The operation is performed on generic object names in the library list (*CURLIB) associated with the job that requested the generic operation.
- *CURLIB: The operation is performed on generic object names in the current library.
- *ALL: The operation is performed on generic object names in all libraries on the system for which you are authorized.
- *USRLIBL: The operation is performed on generic object names in the user part of the library list (*LIBL) for the job.
- *ALLUSR: The operation is performed on all nonsystem libraries (libraries that do not start with the letter Q), except for the following:

  ```
  #CGULIB     #COBLIB     #DFULIB
  #RPGLIB     #SDALIB     #SEULIB
  #DSULIB
  ```

  The operation is also performed on the following Qxxx libraries:

  ```
  QDSNX       QGPL        QGPL38
  QPFRDATA    QRCL        QS36F
  QUSER38     QUSERSYS    QUSRVxRxMx
  ```

  Note: A different library name, of the form QUSRVxRxMx, is added with each release. VxRxMx is the version, release, and modification level of the library.

Additional Rules for Unique Names

Additional rules involving special characters (as an extra character) that apply to the following types of names are:

- A *command label* must be immediately followed by a colon (:). Blanks can follow the colon, but none can precede it. A command label name cannot be a quoted name.
- A CL *variable name* must be preceded by an ampersand (&) to indicate that it is a CL variable used in a CL program.
- A *built-in function name* must be preceded by a percent sign (%) to indicate that it is an IBM-supplied built-in function that can be used in an expression. A built-in function name cannot be a quoted name.

These special characters are not part of the name; each is an additional character attached to a name (making a maximum of 11 characters) indicating to the system what the name identifies.

The names of OS/400 objects, CL program variables, system values, and built-in functions can be specified in the parameters of individual commands as indicated in the syntax diagram for each command. Instead of specifying a constant value, a CL variable name can be used on most parameters in CL programs to specify a value that may change during the running of programs. The names, then, identify which objects and values are used when the command is run.

Identifying OS/400 Objects

Each of the OS/400 objects used by the control language has a name. The object name specified in a CL command identifies which object is used by the operating system to perform the function of the command. For information on generic object names, see "Rules for Specifying Names" earlier in this chapter.

The following sections do not apply to folder and document objects. For information on naming folders and documents, see "Rules for Specifying Folder and Document Names" later in this chapter.

Simple and Qualified Object Names

The name of a specific object can be specified two ways: as a simple name or as a qualified name. A *simple object name* is the name of the object only. A *qualified object name* is the name of the library where the object is stored followed by the name of the object. In a qualified object name, the library name is connected to the object name by a slash (/).

Name Type	Name Syntax	Example
Simple object name	object-name	OBJA
Qualified object name	library-name/object-name	LIB1/OBJB

Either the simple name or the qualified name of an object can be specified if the object exists in one of the libraries named in the job's library list; the library qualifier is optional in this case. A qualified name *must* be specified if the named object is not in a library named in the library list.

Note: A job name also has a qualified form, but it is not a qualified object name because a job is not an OS/400 object. A job name is qualified by a user name and a job number, not by a library name. For more information about the JOB parameter, refer to Appendix A, "Expanded Parameter Descriptions."

Generic Object Names

A *generic object name* may refer to more than one object. That is, a generic name contains one or more characters that are the first group of characters in the names of several objects; the system then searches for all the objects that have those characters at the beginning of their names and are in the libraries named in the library list. A generic name is identified by an asterisk (*) as the last character in the name.

A quoted generic name consists of a generic name enclosed in quotation marks. Unlike quoted names, if there are no special characters between the quotation marks, the quotation marks are not removed. The generic name "ABC*" would cause the system to search for objects whose name begins with *"ABC"*.

A generic name can also be qualified by a library name. If the generic name is qualified, the system searches only the specified library for objects whose names begin with that generic name.

Name Type	Name Syntax	Example
Simple generic name	generic-name*	OBJ*
Qualified generic name	library-name/generic-name*	LIB1/OBJ*
Quoted generic name	"generic-name*"	"ABC*"

OS/400 Object Naming Rules

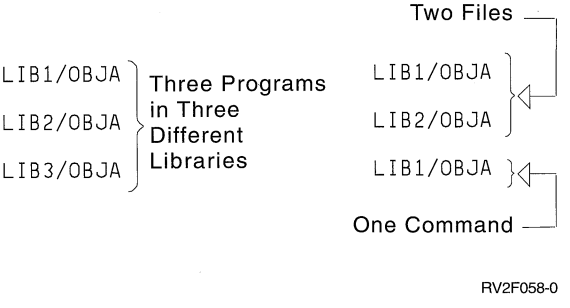
The following rules, in addition to the standard rules for all names, are used to name all OS/400 objects used in control language commands. The syntax diagram for each CL command shows whether a simple object name, a qualified name, or a generic name can be specified.

- *Naming a Single Object*: In the name of a single object, each part (the simple name and the library qualifier name) can have a maximum of 10 characters. For more information on specifying objects, see "Rules for Specifying Names" earlier in this chapter.
- *Naming a User-Created Object*: To distinguish a user-created object from an IBM-supplied object, you should not begin user-created object names with Q because the names of all IBM-supplied objects (except commands) begin with Q. Although you can use as many as 10 characters in CL object names, you may need to use fewer characters to be consistent with the naming rules of the particular high-level language that you are also using. Also, the high-level language might not allow underscores in the naming rules. For example, RPG limits file names to eight characters and does not allow underscores.
- *Naming a Generic Object*: In a generic name, a maximum of nine alphanumeric characters can be used, not including the asterisk (*) that must immediately follow the last character. For more information on specifying objects, see "Rules for Specifying Names" earlier in this chapter.

INV and INV* are valid values where a generic name is accepted. When the name INV is specified, only the object INV is referenced. When the quoted generic name INV* is specified, objects that begin with INV are referred to, such as INV, INVOICE, INVENTORY, and INVENPGM1. When the quoted generic name "INV*" is specified, objects that begin with "INV are referred to, such as "INV%1" and "INV>.

- *Object Library Qualifier Limitations*: No library qualifier can be specified with the object name if the object being created is a library, user profile, line description, controller description, device description, mode description, class-of-service description, or configuration list. A library name can never be qualified because a library cannot be placed in a library. The other object types (*USRPRF, *LIND, *CTLD, *DEVD, *MODD, *COSD, *AUTHLD, *ALTL, and *CFGL) appear to be types that exist only in the QSYS library. When only the name of an object of these object types is accepted, a library qualifier cannot be specified with the object name. On the Display Object Description (DSPOBJD) command, where any object name is accepted, QSYS can be specified.
- *Library List Qualifiers*: The predefined value *LIBL (and others, such as *CURLIB and *ALLUSR) can be used in place of a library name in most commands. *LIBL indicates that the libraries named in the job's library list are used to find the object named in the second part of the qualified name.
- *Duplicate Object Names*: Duplicate names are not allowed for objects of the same type in the same library.

Two objects with the same name cannot be stored in the same library unless their object types are different. Two objects named OBJA can be stored in the library LIBx only if, for example, one of the objects is a program and the other is a file. The following combinations of names and object types could all exist on the system at the same time.



If more than one library contains an object with the same name (and both libraries are in the same library list) and a library qualifier is not specified with the object name, the first object found by that name is used. Therefore, when you

have multiple objects with the same name, you should specify the library name with the object name or ensure that the appropriate library occurs first in the library list. For example, if you are testing and debugging and choose not to qualify the names, ensure that your test library precedes your production library in the library list.

Default Libraries

In a qualified object name, the library name is always optional. If a library name is not specified, the default given in the command's description is used (usually either *CURLIB or *LIBL). If the named object is being created, the current library is the default; when the object is created, it is placed either in the current library or in the QGPL (the general purpose library) if no current library is defined. For objects that already exist, *LIBL is the default for most commands, and the job's library list is used to find the named object. The system searches all of the libraries currently in the library list until it finds the object name specified.

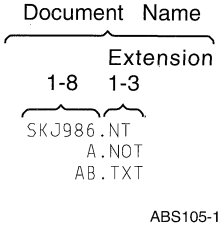
Rules for Specifying Folder and Document Names

Folder names should describe the contents of a folder. The names must be unique and should be easy to type, as well as descriptive to a user. To find a particular folder on the system and change a document stored in it, you must either supply the folder name or select it from a list of names.

Document names should describe the contents of the document. You should give careful consideration to the names you use to help you find the document later. The names must be unique in the folder and should be easy to type, as well as descriptive.

The name you use for a folder or a document must follow these rules:

- The name must be unique within a folder.
- A document or folder name can be 1 to 12 characters long, including an optional extension. If no extension is included, a document or folder name can have a maximum of eight characters. If an extension is included, the extension must start with a period and can have up to three additional characters. An extension in the document name allows you to identify the document by using specific information that can help you do a selective listing of documents on your system.



ABS105-1

- A document or folder name can include any single-byte EBCDIC character *except for the following* special characters that the system uses for other purposes:

Character	Special Uses
Asterisk (*)	Multiplication operator, indicates generic names, and indicates OS/400 reserved values
Slash (/)	Division operator, delimiter within system values, and separates parts of qualified object names
Question Mark (?)	Initiates requests for system help

- Unlike elsewhere in CL, lowercase English letters (a through z) in document and folder names are always treated as though they were entered as uppercase letters (A through Z), even within quoted character strings.
- When a folder is stored in another folder, both folder names are used, separated by a slash (/). That combination of names is called a **folder path**. For example, if a folder named FOLDR2 is stored in FOLDR1, the path for FOLDR2 is FOLDR1/FOLDR2. FOLDR1 is the **first-level folder**. FOLDR2 is the **next-level folder**. The name of a single folder can be 1 to 12 characters long, including an optional extension. A folder path can contain a maximum of 63 characters.

  Folder names should not begin with Q because the system-supplied folder names begin with Q. The following are examples of permitted folder names and folder paths:

```
@LETTERS
FOLDER.PAY
PAYROLL/FOLDER.PAY
#TAX1/FOLD8.TAX/$1988/PAYROLL/FOLDER.PAY
```

Notes:

1. In CL commands, folder path names must be enclosed in apostrophes to prevent the system from processing them as qualified (library/object) names. If an apostrophe is to be part of the name, it must be specified as two consecutive apostrophes.
2. A number of CL commands act on either documents or folders, and some act on both. The abbreviation DLO (document library object) is used when referring to either a document or folder.
3. In CL commands, folder and document names must be enclosed in apostrophes if they contain characters that are CL delimiters.
4. The system does not recognize graphic characters; it recognizes only code points and uses the following assumptions:
   - All folder and document names are encoded using single-byte EBCDIC code pages. Since code points hex 41 through FE represent graphic characters in those code pages, they are the only code points that can be used in folder and document names.

- Code points hex 5C, 61, and 6F represent the asterisk (*), slash (/), and question mark (?) respectively, and cannot be used in folder and document names.
- The code points for lowercase letters in English (hex 81 through 89, 91 through 99, and A2 through A9) are converted to the code points for uppercase letters (C1 through C9, D1 through D9, and E2 through E9, respectively).

More information on code pages that are supported on the AS/400 system is in Chapter 5 of the *Device Configuration Guide* and in the description of the CRTDEVDSP (Create Device Description (Display)) command in this manual.

In addition to the folder and document names previously described, folders and documents are internally classified in the system by their system object names. These are 10-character names derived from date/time stamps, and, while they are generally not known to the user, they may be specified on some CL commands by specifying *SYSOBJNAM for the folder or document name and by specifying the system object name in a separate parameter.

Parameter Values

Parameter values are user-supplied information used during running of a command. An individual value can be specified in any one of these forms:

- Constant (its actual value): The types of constants are character string (includes names), decimal, and logical.
- CL variable name (the name of the variable containing the value): The types of variables are character string (includes names), decimal, and logical. Decimal and logical values must match the type of value expected for the parameter. Character variables can specify any type of value. For example, if a decimal value is expected, it can be specified by a character variable as well as by a decimal variable.
- Expression (the value used is the result of evaluating an expression): The types of expressions are arithmetic, character string, relational, and logical. Expressions can be used as a value for parameters in commands in CL programs only.

A parameter can specify one or a group of such values, depending on the parameter's definition in a command. If a group of values is specified, the parameter is called a *list parameter* because it can contain a list of values.

On commands with key and positional parameters, values can be specified in keyword form, positional form, or a combination of both forms. Parameter values must be enclosed in parentheses if the any of the following conditions are true:

- A keyword precedes the value.
- The value is an expression.
- A list of values is specified.

  Note: If only one value is specified for a list, no parentheses are required.

A description of each type of parameter value is given in the following paragraphs.

Constant Values

A constant is an actual numeric value or a specific character string whose value does not change. Three types of constants can be used by the control language: character string (quoted or unquoted), decimal, and logical.

Character Strings: A *character string* is a string of any EBCDIC characters (alphanumeric and special) that are used as a value. A character string can have two forms: quoted string or unquoted string. Either form of character string can contain as many as 2000 characters.

A *quoted* character string is a string of alphanumeric and special characters that are enclosed in apostrophes. For example, 'Credit limit has been exceeded' is a quoted character string.

The *quoted* string is used for character data that is not valid in an unquoted character string. For example, user-specified text can be entered in several commands to describe the functions of the commands. Those descriptions must be enclosed in apostrophes if they contain more than one word because blanks are not allowed in an unquoted string.

An *unquoted* character string is a string consisting of only alphanumeric characters and the special characters that are shown in the *Unquoted String* column in the figure in the following text. The special characters allow the following to be unquoted character string values:

- Predefined values (* at the beginning)
- Qualified object names (/)
- Generic names (* at the end)
- Decimal constants (+, –, ., and ,)

Any of these unquoted strings can be specified for parameters defined to accept character strings. In addition, some parameters are defined to accept predefined values, names, or decimal values either singly or in combinations.

The following table summarizes the main EBCDIC characters the valid in unquoted and quoted character string values. An X in the last column indicates that the character on the left is valid; refer to the specific notes following the figure that indicate why the character is valid as described.

Name of Character	Character	Unquoted String	Quoted String
Ampersand	&	See Note 5	X
Apostrophe	'	See Note 7	–

Name of Character	Character	Unquoted String	Quoted String
Asterisk (*)	*	See Notes 5, 6	X
At sign	@	X	X
Blank	b		X
Colon	:		X
Comma	,	See Note 1	X
Digits	0-9	See Note 1	X
Dollar sign	$	X	X
Equal	=	See Notes 5, 8	X
Greater than	>	See Notes 5, 8	X
Left parenthesis	(	See Note 4	X
Less than	<	See Notes 5, 8	X
Letters (lowercase)	a-z	See Note 2	X
Letters (uppercase)	A-Z	X	X
Minus	-	See Notes 1, 5	X
Not	¬	See Notes 5, 8	X
Number sign	#	X	X
Percent	%		X
Period	.	See Notes 1, 11	X
Plus	+	See Notes 1, 5	X
Question mark	?		X
Quotation marks	" "	See Note 10	X
Right parenthesis	)	See Note 4	X
Semicolon	;		X
Slash	/	See Notes 3, 5	X
Underscore	_	See Note 9	X
Vertical bar	|	See Notes 5, 8	X

Notes:

1. An unquoted string of all numeric characters, an optional single decimal point (. or ,), and an optional leading sign (+ or -) are valid unquoted strings. Depending on the parameter attributes in the command definition, this unquoted string is treated as either a numeric or character value. On the CALL command or in an expression, this unquoted string is treated as a numeric value; a quoted string is required if character representation is desired. Numeric characters used in any combination with alphanumeric characters are also valid in an unquoted string.
2. In an unquoted string, lowercase letters are translated into uppercase letters.
3. A slash can be used as a connector in qualified names.
4. In an unquoted string, parentheses are valid when used to delimit keyword values and lists, or in expressions to indicate the order of evaluation.
5. In an unquoted string, the characters +, -, *, /, &, |, ¬, <, >, and = are valid by themselves. If they are specified on a parameter that is defined in the command definition with the EXPR(*NO) attribute, they are treated as character values. If they are specified on a parameter that is defined in the command definition with the EXPR(*YES) attribute, they are treated as expression operators.
6. In an unquoted string, the asterisk is valid when followed immediately by a name (such as in a predefined value) and when preceded immediately by a name (such as in a generic name). For further information on unquoted strings in expressions, see Chapter 3, "Expressions in CL Commands."
7. Because an apostrophe within a quoted string is paired with the opening apostrophe (delimiter) and is interpreted as the ending delimiter, an adjacent pair of apostrophes ('') must be used inside a quoted string to represent an apostrophe that is not a delimiter. When characters are counted in a quoted string, a pair of adjacent apostrophes is counted as a single character.
8. In an unquoted string, the characters <, >, =, ¬, and | are valid in some combinations with another character in the same set. Valid combinations are: <=, >=, ¬=, ¬>, ¬<, ||, |<, and |>. If the combination is specified on a parameter that is defined in the command definition with the EXPR(*NO) attribute, it is treated as a character value. If it is specified on a parameter that is defined in the command definition with the EXPR(*YES) attribute, it is treated as an expression operator.
9. In an unquoted string, the underscore is not valid as the first character or when used by itself.
10. Quotation marks are used to delimit a quoted name.
11. A period is valid in a basic name, except as the first character.

The following are examples of quoted string constants:

Constant	Value
'1,2,'	1,2,
'DON''T'	DON'T
'24 12 20'	24 12 20

The following are examples of unquoted strings:

Constant	Meaning
CHICAGO	CHICAGO
FILE1	FILE1
*LIBL	Library list
LIBX/PGMA	Program PGMA in library LIBX
1.2	1.2

Date Values: A date value is a character string that represents a date. Its format is specified by the system value QDATFMT. The length of the date value varies with the format used and whether a separator character is used. For example, if no separator character is used, the length of a date in a Julian format is five characters, and the length of a date in a non-Julian format is six characters. If a separator character is used, the length will be greater. More information on system value QDATFMT is in the *Work Management Guide*.

The system value QDATSEP specifies the optional separator character that can be used when the date is entered. If a separator character is used, the date must be enclosed in apostrophes. For additional information on system value QDATSEP, see the *Work Management Guide*.

A date value can be specified for the parameters of type *DATE. A year value equal to or greater than 40 indicates a year from 1940 through 1999. A year value less than 40 indicates a year from 2000 through 2039. For additional information on parameter value *DATE, see the PARM statement description in Chapter 5, "Command Definition Statements."

Decimal: A decimal value is a numeric string of one or more digits, optionally preceded by a plus (+) or minus (–) sign. A decimal value can contain a maximum of 15 digits, of which no more than nine can follow the decimal point (which can be either a comma or a period). Therefore, a decimal value can have no more than 17 character positions including the plus or minus sign and decimal point (if any). The following are examples of decimal values.

```
123.
  1.23  } Equivalent
  1,23  } Values
 -1,23

    +.017
  6278,954374
-123456.987654321
87654321.123
```

RSLS708-0

Logical Values: A logical value is a single character (1 or 0) enclosed in apostrophes. It is often used as a switch to represent a condition such as on or off, yes or no, and true or false. When used in expressions, it can be optionally preceded by *NOT or ¬. The following are examples of logical values:

Constant	Value	Meaning
'0'	0	Off, no, or false
'1'	1	On, yes, or true

Hexadecimal Values: A hexadecimal value is a constant made up of a combination of the hexadecimal digits A through F and 0 through 9. All character strings except names, dates, and times can be specified in hexadecimal form. To specify a hexadecimal value, the digits must be specified in multiples of two, be enclosed in apostrophes, and be preceded by an X. Examples are: X'F6' and X'A3FE'.

Note: Care should be used when entering hexadecimal values in the range of 00 through 3F, or the value FF. If these characters are shown or printed, they may be treated as device control characters producing unpredictable results.

Floating-Point Constants: A floating-point constant is a representation of a number that consists of:

- A significand sign: The significand sign may be + or –. The significand sign is optional; it is assumed to be + if no sign is specified.
- A significand: The significand must contain a decimal point. The maximum number of digits that can be specified for the significand is 253; however, only the first 17 significant digits are used.
- An exponent character: The exponent character must be E.
- An exponent sign: The exponent sign must be + or –. The significand sign is optional; it is assumed to be + if no sign is specified.
- An exponent: The exponent must be an integer; numbers 0 through 9 are valid. The maximum number of digits that can be specified is three.

All floating-point constants are stored as double-precision values. No blanks are allowed between any of the parts of a floating-point constant, and the parts must be in the order listed above.

Three commands have parameters for which floating-point constants can be specified:

- Call Program (CALL) command: You can use the PARM parameter to pass a floating-point constant to a called program. Any program you call must receive a floating-point constant as a double precision value.
- Change Program Variable (CHGPGMVAR) command: You can use the VALUE parameter to change a floating-point variable in a program.
- Copy File (CPYF) command: You can use floating-point construction in the FROMKEY, TOKEY, and INCREL parameters to select which records are copied from a database file.

More information on floating-point constants is in the *DDS Reference* manual.

Variables

A *variable* contains a data value that can be changed when a program is run. The variable is used in a command to pass the value that it contains at the time the command is run. The change in value can result if one of the following conditions occur: the value is received from a data area, a display device file field, or a message; the value is passed as a parameter; a Change Variable (CHGVAR) command is run in the program; or another program that is called changes the value before returning it.

The variable name identifies a value to be used; the name points to where the actual data value is. Because CL variables are valid only in CL programs, they are often called *CL program variables* or, simply, CL variables. CL variable names must begin with an ampersand (&).

CL variables can be used to specify values for almost all parameters of CL commands. When a CL variable is specified as a parameter value and the command containing it is run, the current value of the variable is used as the parameter value. That is, the variable value is passed as if the user had specified the value as a constant.

Because it is generally true that CL variables can be used for most parameters of commands in CL programs, the

command descriptions usually do not mention CL variables. For parameters that are restricted to constants only (such as in the DCL command), to CL variables only (such as all of the parameters of the Retrieve Job Attributes (RTVJOBA) command), or to specific types of variables (such as on the RTVJOBA or Retrieve Message (RTVMSG) command), the individual parameter descriptions specify those limitations. Otherwise, if the command is allowed in a CL program, CL variables can be used in place of a value, even with parameters that accept only predefined values. For example, a SAVE parameter having only predefined values of *YES and *NO can have a CL variable specified instead; its value can then be changed to *YES or *NO, depending on its value when the command is run.

A CL variable must contain only one value; it may not contain a list of values separated by blanks.

The value of any CL program variable can be defined as one of the following types:

- Character: A character string that can contain a maximum of 9999 characters. The character string can be coded in quoted or unquoted form, but only the characters in the string itself are stored in the variable.
- Decimal: A packed decimal value that can contain a maximum of 15 digits, of which no more than nine can be decimal positions.
- Logical: A logical value of '1' or '0' that represents on/off, true/false, or yes/no.

If value is:	CL variable can be declared as:
Name	Character
Date or time	Character
Character string	Character
Numeric	Decimal or character
Logical	Logical or character

Expressions

An expression is a group of constants or variables, separated by operators, that produces a single value. The operators specify how the values are combined to produce the single value or result. The operators can be arithmetic, character string, relational, or logical. The constants or variables can be character, decimal, or logical. For example, the expression (&A + 1) specifies that the result of adding 1 to the value in the variable &A is used in place of the expression.

Character string expressions can be used in certain command parameters defined with EXPR(*YES) in CL programs. An expression can contain the built-in functions %SUBSTRING (or %SST) and %SWITCH, which are covered in detail in Chapter 3, "Expressions in CL Commands." The types of expressions and examples of each are described there.

Lists of Values

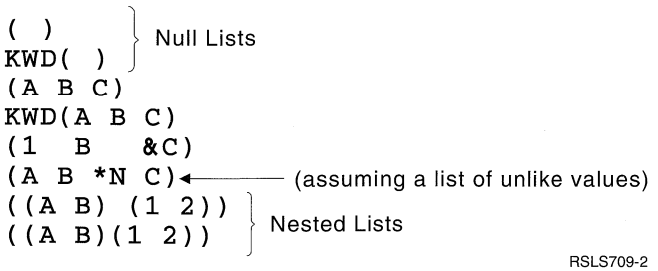
A list of values is one or more values that can be specified for a parameter. Not all parameters can accept a list of values. A *list parameter* can be defined to accept a specific set of multiple values that can be of one or more types. Values in the list must be separated by one or more blanks. Each list of values is enclosed by parentheses, indicating that the list is treated as a single parameter. Parentheses are used even when a parameter is specified in positional form. To determine whether a list can be specified for a parameter, and what kind of list it can be, refer to the parameter description under the appropriate command description.

A list parameter can be defined to accept a list of multiple like values (a simple list) or a list of multiple unlike values (a mixed list). Each value in either kind of list is called a *list element.* List elements can be constants, variables, or other lists; expressions are not allowed.

- A *simple list* parameter accepts one or more values of the type allowed by a parameter. For example, (RSMITH BJONES TBROWN) is a simple list of three user names.
- A *mixed list* parameter accepts a fixed set of separately defined values that are in a specific order. Each value can be defined with specific characteristics such as type and range. For example, LEN(5 2) is a mixed list in which the first element (5) gives the length of a field and the second element (2) gives the number of decimal positions in that field.
- For many parameters defined to accept lists, predefined single values can be specified in place of a list of values. One of these single values can be the default value, which can be either specified or assumed if no list is specified for a simple or mixed list. To determine what defaults are accepted for a given list parameter, refer to the description of the parameter in the description of the command for which the parameter is defined and used.

  Note: *N cannot be specified in a simple list, but it can be specified in a mixed list. Also, individual parameters passed on the CALL command cannot be lists.
- The maximum level of nesting of lists inside lists is three, including the first. These are indicated by three nested levels of parentheses.

The following are examples of lists:



The last two examples contain two lists nested inside a list: the first list contains values of A and B, and the second list contains values of 1 and 2. The space between the two nested lists is not required. Blanks are the separators between the values inside each nested list, and the sets of parentheses group the nested values into larger lists.

Syntax Coding Rules (Summary)

This section contains a summary of general information needed to properly code CL commands.

Delimiters

- Blanks are the basic separators between the parts of a command:
  - Between command label and command name (not required, because the colon [:] is the delimiter).
  - Between command name and first parameter, and between parameters.
  - Between values in a list of values (not required between ending and beginning parentheses that enclose nested lists inside a list).
  - Between the slashes and the name or label of some job control commands, like // DATA (not required).
- Blanks *cannot* separate a parameter's keyword from the left parenthesis preceding its values. When a keyword is used, parentheses must be used to enclose the values; blanks *can* occur between the parentheses and the values. For example, KWD(A) is valid.
- Multiple blanks are treated as a single blank, unless they occur in a quoted string or a comment.
- A colon must immediately follow a command label. Only one label can be used on any command (LABEL1: DCLF).
- Apostrophes must be used to specify the beginning and end of a quoted character string. If a character string contains special characters, such as blanks, apostrophes are required. If an apostrophe must be used in the quoted string, two apostrophes must be entered consecutively to indicate that it is an apostrophe and not the end of the quoted string.
- Parentheses must be used:
  - On parameters that are specified (coded) in keyword form
  - To group multiple values in a single list, in a positional parameter, or around expressions
  - To indicate a list (of none, one, or several elements) nested inside *another* list
- Sets of parentheses inside parentheses can be entered as long as they are paired, up to the maximum of five nested levels in logical expressions or three nested levels in lists of values.
- Comments can appear wherever blanks are permitted, except after a continuation character on the same line or record.
- A plus or minus sign at the end of a line indicates that the command is continued on the following line. Blanks following a + or – sign in the same record are ignored; blanks in the next record that precede the first nonblank character are ignored when + is specified and included when – is specified. One blank must precede the + sign when it is used between separate parameters or values.

Parameters

- All required parameters must be coded.
- If an optional parameter is not coded, the system uses its default value if the parameter has one. In the syntax diagram of each command, default values are indicated by an arrow that points to them. If no default value is indicated, the default varies (depending on other parameter values) and is described in the text, or the action taken does not require that parameter.
- Words or abbreviations specified in capital letters in the command and parameter descriptions must be coded as shown. This is true of all command names, the keywords of parameters (if used), and many parameter values. Lowercase letters coded not in quoted strings or comments are translated to uppercase letters.
- The values for the key parameters (if applicable) must be entered on the prompt before the remaining parameters will be shown. Parameters preceding the letter ***K*** on the syntax diagram are key parameters.

  If the key parameter symbol is absent, no key parameters exist for the command.
- Parameters may not be coded in positional form past the letter ***P*** found in the syntax diagrams (if applicable).

  If the positional coding limit symbol is absent, all parameters in the command may be coded in positional form. The order of positional coding is the order in which the parameters are presented in the syntax diagram and in the command description.

Values

- The first character in all names must be an alphabetic character (A through Z, $, #, @, or a double quotation mark (")). Names must not exceed 10 characters (CL variable names and built-in function names can have 11 characters maximum, including the preceding & or % characters). In some commands, the names of objects

can be specified in qualified form (library-name/object-name).

- Predefined values that begin with an asterisk can be used only for the purposes intended, unless they are included in comments or quoted strings. These include predefined parameter values (*ALL, for example), symbolic operators (*EQ, for example), and the null value (*N).
- In a CL program, a variable can be specified for all parameters, except where that is explicitly restricted. The user-specified value of the variable is passed as if it had been specified on the command.
- In a CL program, a character string expression can be specified for any parameter defined with EXPR(*YES). The resulting value of the expression is passed as if the value had been specified on the command.
- Null (omitted) values are specified with the characters *N, which mean that no value was specified and the default value, if one exists, should be used. *N is needed only when another value following the omitted value is being specified as a positional parameter or an element in a list.
- Either a comma or a period can be used to indicate a decimal point in a numeric value. The decimal point is the only special character allowed between digits in the numeric string; there is no delimiter for indicating thousands.
- When repetition is indicated for a parameter:
  - A predefined value is not coded more than once in a series of values.
  - As many user-defined values (like names or numeric limits) can be entered as there are different values or names, up to the maximum number of repetitions allowed.

Note: When you are using parameters that have the same name in different commands, the meaning of (and the values for) that parameter in each command may be somewhat different. Refer to the correct command description for the explanation of the parameter you are using. For some parameters, you can also refer to Appendix A, "Expanded Parameter Descriptions" for both general information about a parameter and an expanded description of its values coded in commands.

Chapter 3. Expressions in CL Commands

A character string expression can be used for any parameter, element, or qualifier defined with EXPR(*YES) in the command definition object. Any expression can be used as a single parameter in the Change Variable (CHGVAR) and If (IF) commands. An expression in its simple form is a single constant, a variable, or a built-in function. An expression usually contains two operands and an operator that indicates how the expression is to be evaluated. Two or more expressions can be combined to make a complex expression.

The following types of expressions are supported in CL programs:

- Arithmetic (`&VAR + 15`)
- Character string (`SIX || TEEN`)
- Logical (`&VAR & &TEST`)
- Relational (`&VAR > 15`)

Each type is discussed on the following pages.

A *complex* expression contains multiple operands, operators that indicate what operation is performed on the operands, and parentheses to group them. Only one operator is allowed between operands, except for the + and - signs when they immediately precede a decimal value (as a signed value), and the *NOT operator when it is used in a logical expression.

No complex expression can have more than five nested levels of parentheses, including the outermost (required) level.

Arithmetic and character string expressions can be used together in a complex expression if they are used with relational and logical operators; for example: `(A=B&(1+2)=3)`. A pair of arithmetic expressions or a pair of character string expressions can be compared within a relational expression. Also, relational expressions can be used within a logical expression.

Operators in Expressions

Operators are used in expressions to indicate an action to be performed on the operands in the expression or the relationship between the operands. There are four kinds of operators, one for each of the four types of expressions:

- Arithmetic operators (`+, -, *, /`)
- Character operator (`||, |>, |<`)
- Logical operators (`&, |, ¬`)
- Relational operators (`=, >, <, >=, <=, ¬=, ¬>, ¬<`)

Each operator must be between the operands of the expression in which it is used; for example, `(&A + 4)`. Operators can be specified as a predefined value (for example, *EQ) or as a symbol (for example, =).

- All predefined value operators must have a blank on each side of the operator:

  `(&VAR *EQ 7)`.

- Except for the division operator (/), symbolic operators need no blanks on either side. For example, either `(&VAR=7)` or `(&VAR = 7)` is valid.

  Where the division operator *follows* a variable name, the division operator must be preceded by a blank. For example, `(&VAR / 5)` or `(&VAR /5)` is valid; `(&VAR/5)` is not valid.

The following character combinations are the predefined values and symbols that represent the four kinds of operators; they should not be used in unquoted strings for any other purpose. For further information, refer to "Operators in Expressions."

Predefined Value	Predefined Symbol	Meaning	Type
	+	Addition	Arithmetic operator
	–	Subtraction	Arithmetic operator
	*	Multiplication	Arithmetic operator
	/	Division	Arithmetic operator
*CAT	||[1]	Concatenation	Character string operator
*BCAT	|>[1]	Blank insertion with concatenation	Character string operator
*TCAT	|<[1]	Blank truncation with concatenation	Character string operator
*AND	&	AND	Logical operator
*OR	|[1]	OR	Logical operator
*NOT	¬[2]	NOT	Logical operator
*EQ	=	Equal	Relational operator
*GT	>	Greater than	Relational operator
*LT	<	Less than	Relational operator
*GE	≥	Greater than or equal	Relational operator
*LE	≤	Less than or equal	Relational operator
*NE	¬=[2]	Not equal	Relational operator
*NG	¬>[2]	Not greater than	Relational operator

Predefined Value	Predefined Symbol	Meaning	Type
*NL	¬<[2]	Not less than	Relational operator

Notes:

1 In some national character sets and in the multinational character set, the character | (hexadecimal 4F) is replaced by the character ! (exclamation point). Either ! or *OR can be used as the OR operator and either !! or *CAT, !> or *BCAT, and !< or *TCAT can be used for concatenation in those character sets.

2 In some national character sets and in the multinational character set, the character ¬ (hexadecimal 5F) is replaced by the character ┐. Either ┐ or *NOT can be used as the NOT operator in those character sets.

Priority of Operators When Evaluating Expressions

When multiple operators occur in an expression, the expression is evaluated in a specific order depending upon the operators in the expression. Parentheses can be used to change the order of expression evaluation. The following table shows the priority of all the operators used in expressions, including signed decimal values.

Priority	Operators
1	signed (+ and -) decimal values, *NOT, ¬
2	* , /
3	+, - (when used between two operands)
4	*CAT, ||, *BCAT, |>, *TCAT, |<
5	*GT, *LT, *EQ, *GE, *LE, *NE, *NG, *NL, >, <, =, >=, <=, ¬=, ¬>, ¬<
6	*AND, &
7	*OR, |

A priority of 1 is the highest priority (signed values are evaluated first); a priority of 7 is the lowest priority (OR relationships are evaluated last). When operators with different priority levels appear in an expression, operations are performed according to priorities.

When operators of the *same* priority appear in an expression, operations are performed from left to right within the expression. Parentheses can always be used to control the order in which operations are performed. The value of a parenthetical expression is determined from the innermost level to the outermost level, following the priorities stated above within matching sets of parentheses.

Arithmetic Expressions

The operands in an arithmetic expression must be decimal constants or decimal CL variables. An arithmetic operator (only in symbolic form) must be between the operands. The results of all arithmetic expressions are decimal values, which may be stored in a CL variable.

Note: The division operator (/) must be preceded by a blank if the operand that precedes it is a variable name. (For example, `&A /2`, *not* `&A/2`.) All other arithmetic operators may optionally be preceded or followed by a blank.

Arithmetic operands can be signed or unsigned; that is, each operand (whether it is a numeric constant or a decimal CL variable) can be immediately preceded by a plus (+) or minus (-) sign, but a sign is not required. When used as a sign, no blanks can occur between the + or - and its value. For example, a decimal constant of 23.7 can be expressed as +23.7 or -23.7 (signed) or as 23.7 (unsigned).

The following are examples of arithmetic expressions:

```
(&A + 1)                (&A + &B -15)
(&A - &F)               (&A+&B-15)
(&A + (-&B))
```

If the last nonblank character on a line is a + or -, it is treated as a continuation character and not as an arithmetic operator.

Character String Expressions

The operands in a character string expression must be quoted or unquoted character strings, character variables, or the substring (%SUBSTRING or %SST) built-in function. The value associated with each variable or built-in function must be a character string. The result of concatenation is a character string.

There are three operators that can be used in character string expressions. They are:

- *CAT (concatenation, symbol ||)
- *BCAT (concatenation with blank insertion, symbol |>)
- *TCAT (concatenation with trailing blank truncation, symbol |<)

These operators concatenate (or join) two character strings, but each has a slightly different function.

The *CAT Operator: The *CAT operator concatenates two character strings. For example:

```
ABC *CAT DEF becomes ABCDEF
```

Blanks are included in the concatenation. For example:

```
'ABC ' *CAT 'DEF ' becomes 'ABC DEF '
```

The *BCAT Operator: The *BCAT operator truncates all trailing blanks in the first character string; one blank is inserted, then the two character strings are concatenated. All leading blanks on the second operand are not truncated. For example:

```
ABC *BCAT DEF becomes ABC DEF
'ABC  ' *BCAT DEF becomes 'ABC DEF'
```

The *TCAT Operator: The *TCAT operator truncates all trailing blanks in the first character string, then the two character strings are concatenated. All leading blanks on the second operand are not truncated. For example:

```
ABC *TCAT DEF becomes ABCDEF
'ABC ' *TCAT DEF becomes 'ABCDEF'
ABC *TCAT ' DEF' becomes 'ABC DEF'
'ABC '*TCAT ' DEF' becomes 'ABC DEF'
```

All blanks that surround the concatenation operator are ignored, but at least one blank must be on each side of the reserved value operator (*CAT, *BCAT, or *TCAT). If multiple blanks are wanted in the expression, a quoted character string (a character string enclosed within apostrophes) must be used.

Examples of Character String Expressions: The following are examples of string expressions. Assume the following variables:

Variable	Value
&AA	'GOOD '
&BB	'REPLACEMENT'
&CC	'ALSO GOOD'
&DD	'METHOD'

Expression	Result
(&AA || &BB)	GOOD REPLACEMENT
(&AA||&BB)	GOOD REPLACEMENT
(&AA *CAT &BB)	GOOD REPLACEMENT
(&CC |> &DD)	ALSO GOOD METHOD
(&CC *BCAT &DD)	ALSO GOOD METHOD
(A *CAT MOUSE)	AMOUSE
('A ' *CAT MOUSE)	A MOUSE
(FAST *CAT MOUSE)	FASTMOUSE
('FAST ' *BCAT MOUSE)	FAST MOUSE
('FAST ' *TCAT MOUSE)	FASTMOUSE
('AB' *CAT 'CD')	ABCD
('AB' *BCAT 'CD')	AB CD
('AB' *TCAT 'CD')	ABCD
(%SST(&AA 1 5) *CAT (%SST(&BB 3 5))	GOOD PLACE
(%SST(&CC 1 9) *BCAT (%SST(&BB 3 5))	ALSO GOOD PLACE
(&AA *CAT ' TIME')	GOOD TIME
(&CC *BCAT TIME)	ALSO GOOD TIME

Another Example Using Character Strings and Variables: The following example shows how several character variables and character strings can be concatenated to produce a message for a work station operator. The example assumes that the variables &DAYS and &CUSNUM were declared as character variables, not decimal variables.

```
DCL      VAR(&MSG) TYPE(*CHAR)          LEN(100)
          •
          •
CHGVAR        &MSG ('Customer' *BCAT &CUSNAME +
                    *BCAT 'Account Number' *BCAT +
                    &CUSNUM *BCAT 'is overdue by' +
                    *BCAT &DAYS *BCAT 'days.')
```

After the appropriate variables have been substituted, the resulting message might be:

```
Customer ABC COMPANY Account Number 12345
is overdue by 4 days.
```

If the variables &DAYS and &CUSNUM had been declared as decimal variables, two other CHGVAR commands would have to change the decimal variables to character variables before the concatenation could be performed. If, for example, two character variables named &DAYSALPH and &CUSNUMALPH were also declared in the program, the CHGVAR commands would be:

```
CHGVAR  &DAYSALPH  &DAYS

CHGVAR  &CUSNUMALPH  &CUSNUM
```

Then instead of &DAYS and &CUSNUM, the new variables &DAYSALPH and &CUSNUMALPH would be specified in the CHGVAR command used to concatenate all the variables and character strings for &MSG.

Relational Expressions

The operands in a relational expression can be arithmetic or character string expressions; they can also be logical constants and logical variables. Only two operands can be used with each relational operator. The data type (arithmetic, character string, or logical) must be the same for the pair of operands. The result of a relational expression is a logical value '0' or '1'.

Refer to the table under "Operators in Expressions" for the meanings of the relational operators, which can be specified

by symbols (=, >, <, >=, <=, ¬=, ¬>, ¬<) or their reserved values (*EQ, *GT, *LT, *GE, *LE, *NE, *NG, *NL).

If an operation involves character fields of unequal length, the shorter field is extended by blanks added to the right.

Arithmetic fields are compared algebraically; character fields are compared according to the EBCDIC collating sequence.

When logical fields are compared, a logical one ('1') is greater than logical zero ('0'). Symbolically, this is ('1' > '0').

The following are examples of relational expressions:

```
(&X *GT 25)
(&X > 25)
(&X>25)

(&NAME *EQ GSD)
(&NAME *EQ &GSD)
(&NAME *EQ 'GSD')
(&BLANK *EQ ' ')
```

Logical Expressions

The operands in a logical expression consist of relational expressions, logical variables, or constants, separated by logical operators. Two or more of these types of operands can be used in combinations, making up two or more expressions within expressions, up to the maximum of five nested levels of parentheses. The result of a logical expression is a '0' or '1' that can be used as part of another expression or saved in logical variables.

The logical operators used to specify the relationship between the operands are *AND and *OR (as reserved values), and & and | (as symbols). The AND operator indicates that both operands (on either side of the operator) have to be a certain value to produce a particular result. The OR operator indicates that one or the other of its operands can determine the result.

The logical operator *NOT (or ¬) is used to negate logical variables or logical constants. All *NOT operators are evaluated before the *AND or *OR operators are evaluated. All operands that follow *NOT operators are evaluated before the logical relationship between the operands is evaluated.

The following are examples of logical expressions:

```
((&C *LT 1) *AND (&TIME *GT 1430))
(&C *LT 1 *AND &TIME *GT 1430)
((&C < 1) & (&TIME *GT 1430))
((&C<1)&(&TIME>1430))

(&A *OR *NOT &B)
(&TOWN *EQ CHICAGO *AND &ZIP *EQ 60605)
```

Two examples of logical expressions used in the IF command are:

```
IF &A CALL PROG1
IF (&A *OR &B) CALL PROG1
```

%BINARY Built-In Function

The binary (%BINARY) built-in function operates on a character string that is contained in a CL character variable.

%BINARY or %BIN can be used in expressions and as either operand (receiver) of the Change Variable (CHGVAR) command. See the CHGVAR command description for more information.

Note: The binary built-in function can also be used on command parameters that are defined as numeric (*DEC, *INT2 and *INT4) and EXPR(*YES) has been specified.

See "Built-In Function Syntax Diagrams" on page P1-26 for the %BINARY syntax diagram.

The binary built-in function treats the contents of the specified CL character variable, starting at the position specified for a length of 2 or 4 characters, as a signed binary integer.

When the binary built-in function is used with the VAR parameter on the CHGVAR command, the decimal number or arithmetic expression in the VALUE parameter is converted to a 2-byte or 4-byte signed binary integer. A decimal fraction is not included.

If the starting position and length are not specified, then a starting position of 1 and the length of the character variable specified is used. The length of the character variable must be declared as 2 or 4.

The following are examples of how the %BINARY built-in function can be used.

Example 1: Converting binary to decimal

```
DCL  VAR(&N)  TYPE(*DEC)  LEN(3 0)
DCL  VAR(&B2)  TYPE(*CHAR)  LEN(2)  VALUE(X'0012')
CHGVAR  &N  %BINARY(&B2)
```

The content of character variable &B2 is treated as a 2-byte signed binary number and is converted to its decimal equivalent of 18. It is then assigned to the decimal variable &N.

Example 2: Converting decimal to binary

```
CHGVAR  %BIN(&B2)  &N
```

The number contained in the decimal variable &N is converted to a 2-byte signed binary number and is placed in the first and second bytes of the character variable &B2.

Example 3: Used within an arithmetic expression

```
CHGVAR  &N  VALUE(%BIN(&B2) + 4)
```

The contents of character variable &B2 is treated as a 2-byte signed binary integer and is converted to its decimal equivalent of 18. The decimal number 4 is then added and the sum, 22, is assigned to the decimal variable &N.

Example 4: Converting decimal to binary with truncation

```
CHGVAR  &BINARY(&B2)  VALUE(122.567)
```

The number 122.567 is truncated to the whole number 122 and is then converted to a 2-byte signed binary integer and assigned to the character variable &B2. Character variable &B2 will then contain the hexadecimal equivalent of X'007A'.

Example 5: Converting a negative number

```
DCL  VAR(&B4)  TYPE(*CHAR)  LEN(4)
  CHGVAR  &BIN(&B4)  VALUE(-45)
```

The value -45 is converted to a 4-byte signed binary integer assigned to the character variable &B4. Character variable &B4 then contains the hexadecimal equivalent of X'FFFFFFD3'.

Example 6: Used on the IF command

```
IF  COND(&BIN(&B4) *EQ 0)  THEN(GOTO ENDIT)
```

The content of character variable &B4 is treated as a 4-byte signed binary integer and is compared to the decimal number 0. If they are equal, the command following the label ENDIT is run. If they are not equal, the command following the IF command is run.

Example 7: Varying length character string to CPP

```
PGM  PARM(&P ... )

  DCL  VAR(&P)  TYPE(*CHAR)  LEN(202)

  DCL  VAR(&L)  TYPE(*DEC)  LEN(5 0)
  DCL  VAR(&C)  TYPE(*CHAR)  LEN(200)
       •
       •
       •
  CHGVAR  &L  %BINARY(&P 1 2)
  CHGVAR  &C  %SST(&P 3 &L)
       •
       •
       •
ENDPGM
```

This program is the command processing program CPP for a command with a first parameter defined with the attributes TYPE(*CHAR), LEN(200) and VARY(*YES). The first two bytes of character variable &P contain the length of the parameter as *INT2, a 2-byte signed binary integer. The character string specified on the command starts in position 3 of the variable &P. The maximum length of the character string is 200 characters.

The first CHGVAR command retrieves the length from the first two character positions of variable &P and treats the 2 bytes as a signed binary integer. The bytes are converted to the decimal equivalent of the signed binary integer, and are assigned to the decimal variable &L.

The second CHGVAR command retrieves the contents of the parameter by making variable &P a substring and assigning it to variable &C.

%SUBSTRING Built-In Function

The substring built-in function operates on a character string that is contained in a CL character variable or in a local data area. %SUBSTRING or %SST can be used in expressions and as either operand (receiver) of the Change Variable (CHGVAR) command. For more information, see the description of the CHGVAR command. This built-in function can be coded as either %SUBSTRING or %SST.

See "Built-In Function Syntax Diagrams" on page P1-26 for the %SUBSTRING syntax diagram.

This built-in function produces a substring from the contents of the specified CL character variable or local data area. The substring begins at the specified starting position in the value and continues for the length specified. For example:

```
%SST(&TEST  5  3)
```

In this example, a portion of the variable &TEST is referenced. That position (or substring) is 3 characters long and begins with the fifth character position. If &TEST contains ABCDEFGHIJ, the resulting substring will be EFG.

CL variables can also be used to specify the starting position and the length values in the function. For example:

```
CHGVAR   &X  %SST(*LDA  &B  &C)
```

The value of the character variable named &X is to be replaced by the value in the job's local data area, starting at the position obtained from variable &B and continuing for the length specified by the value in &C.

```
RTVJOBA   SWS(&JOBSWS)

CHGVAR   VAR(&CURSW4)  VALUE(%SST(&JOBSWS  4  1))
```

In this example, the Retrieve Job Attributes (RTVJOBA) command is used to retrieve the current value of the job's eight job switches. The CHGVAR command is then used to extract the current value of the fourth job switch only and store it in the variable &CURSW4. If the value of the eight job switches retrieved in &JOBSWS is 10010000, the second 1 would be stored in &CURSW4.

%SWITCH Built-In Function

The built-in function %SWITCH tests one or more of the eight job switches in the current job and returns a logical value of 1 or 0. If every job switch tested by %SWITCH has the value indicated, the result is a 1 (true); if any switch tested does not have the value indicated, the result is a 0 (false).

The 8-character mask is used to indicate which job switches are tested, and what value each switch is tested for. Each position in the mask corresponds with one of the eight job switches in a job. Position 1 corresponds with job switch 1, position 2 with switch 2, and so on. Each position in the mask can be specified as one of three values: 0, 1, or X.

0 The corresponding job switch is tested for a 0 (off).
1 The corresponding job switch is tested for a 1 (on).
X The corresponding job switch is not tested. The value in the switch does not affect the result of %SWITCH.

See "Built-In Function Syntax Diagrams" on page P1-26 for the %SWITCH syntax diagram.

If %SWITCH(0X111XX0) is specified, job switches 1 and 8 are tested for 0s, switches 3, 4, and 5 are tested for 1s, and switches 2, 6, and 7 are not tested. If each job switch contains the value (1 or 0 only) shown in the mask, the result of %SWITCH is true (1).

Function %SWITCH can be used in the Change Variable (CHGVAR) and If (IF) commands. On the CHGVAR command, it can be used in place of a logical variable in the VALUE parameter. On the IF command, it can be used in the COND parameter as the logical expression to be tested.

The following two examples show how the same mask can be used to control a branch in a program (the IF command), or to set the value of a variable (the CHGVAR command).

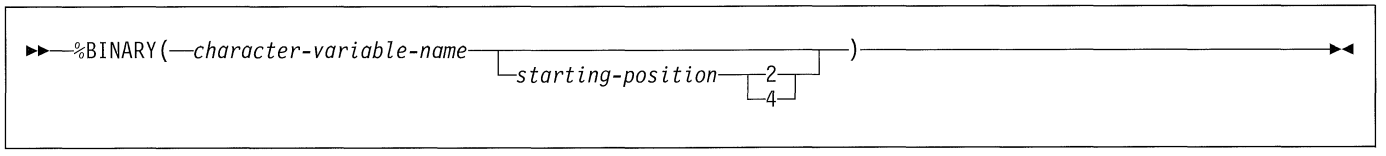
```
IF   COND(%SWITCH(0X111XX0))  THEN(GOTO C)

CHGVAR   VAR(&A)  VALUE(%SWITCH(0X111XX0))
```

If job switches 1, 3, 4, 5, and 8 respectively contain 0, 1, 1, 1, and 0 respectively when %SWITCH(0X111XX0) is specified in the IF command, the result is true and the program branches to the command having label C. If one or more of the switches tested do not have the values indicated in the mask, the result is false and the branch does not occur. If the same mask is used in the CHGVAR command and the result is true, the variable &A is set to a '1'; if the result is false, &A is set to a '0'. Note that &A must be declared as a logical variable.

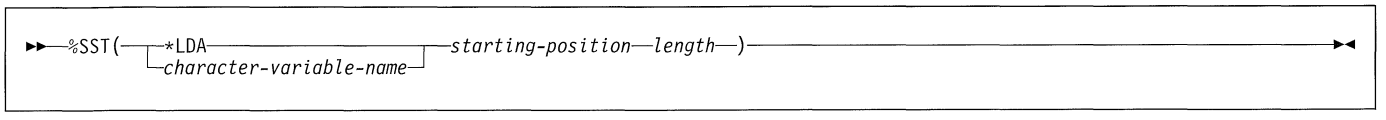
Built-In Function Syntax Diagrams

The syntax diagrams of the %BINARY, %SUBSTRING, and %SWITCH built-in functions appear in the following examples.

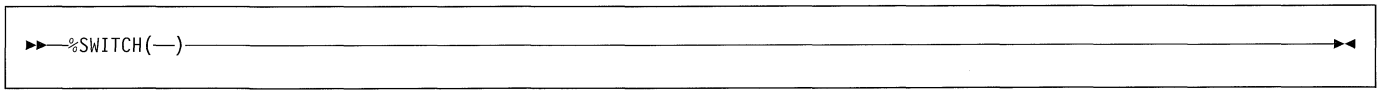
The syntax of the binary built-in function is:



The syntax of the substring built-in function is:



The syntax of the switch built-in function is:



Chapter 4. Format of Command Descriptions

First, a syntax diagram presents all the parameters and values that can be coded on the command. Each command follows the same format. Next, the general description of the command briefly explains the function of the command.

It should be noted that, because a command is an AS/400 object, each command can be authorized for specific users or authorized for use by the public (all users authorized in some way to use the system). Because this is true for nearly every command, it is not stated in each command description. The *Security Reference*, contains additional information about IBM-supplied user profiles and the commands first authorized for each one.

How Commands Are Described

Each command description follows the same format. First, a syntax diagram presents all the parameters and values that can be coded on the command, and it includes special syntax dependencies and parameter restrictions. Next, the purpose of the command and restrictions on its use are described. Then, each parameter and its choice of values are described. Finally, coded examples of the command are given. For some commands, additional information is supplied following the examples.

Command Syntax

The command syntax is presented in the syntax diagram. The syntax diagram shows all the parameters and values that are valid for the command. The parameters are divided into two groups: those that must be coded (required), and those that need not be coded (optional). Arrows pointing to the right indicate default values. These default values are used by the system for parameters or parts of parameters that are not coded.

A complete description of the syntax diagram is provided later in this chapter under "How to Interpret Syntax Diagrams."

Command Description

The general description of the command (under the heading "Purpose") briefly explains the function of the command and any relationships it has with a program or with other commands. If there are restrictions on the use of the command, they are described under the heading "Restrictions."

Parameter Descriptions

Parameter descriptions in the text are presented in the same order as in the syntax diagram. The syntax diagram shows the order in which the parameters must be specified if the values are specified positionally (that is, without keywords). If a parameter has more than one value, the values are described in the same order as shown in the syntax diagram. The default value, if there is one, is always first and is shown as an underlined heading at the beginning of the text that describes the value.

The description of each parameter explains what the parameter means, what it specifies, and the dependent relationships it has with other parameters in the command. When the parameter has more than one value, the information that applies to the parameter as a whole is covered first, then the specific information for each of the values is described after the name of each value.

Command Coding Examples

Each command description shows at least one coded example. Where necessary, several examples are provided for commands with many parameters and several logical combinations.

For clarity, each example is coded in keyword form only. The same examples could be coded either in positional form or in a combination of keyword and positional forms.

Additional Command Considerations

A section called "Additional Considerations" follows the coded examples of some commands to present additional useful information about the command.

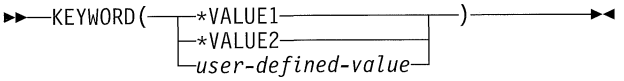
How to Interpret Syntax Diagrams

Syntax diagrams for each command show how you enter the command and its parameters and values in programs, in batch, or interactively on the command line. "How to Read Syntax Diagrams" briefly describes the basic rules you need to read syntax diagrams. "How Syntax Diagrams Show Command Structure" describes in more detail how syntax diagrams reflect CL command structure.

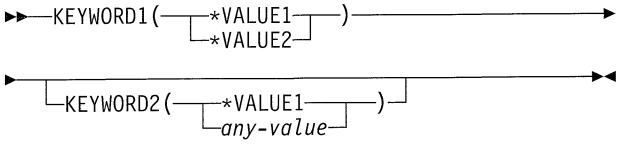
How to Read Syntax Diagrams

Syntax diagrams show the parameters and values used by each CL command. Each syntax diagram specifies, for one command, the parameters that can be coded in the command and the choice of values that are valid for each parameter. Refer to "How Syntax Diagrams Show Command Structure" for a more detailed description of syntax diagrams and how they reflect command structure.

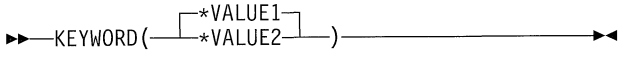
- **Start** here ▸▸— and end here —▸◂.
- **Parameter keywords, predefined values** (values beginning with * or Q), and **user-defined values** must be entered with parentheses as shown below:



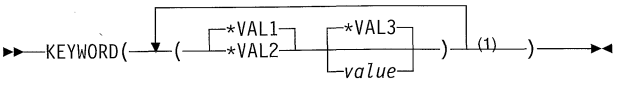
- **Required parameters** must be specified. **Optional parameters** do not have to be entered. In this sample, you must enter KEYWORD1 with a value, but you do not have to enter KEYWORD2.



- **Default values** do not have to be entered. They are used when you do not specify a parameter. In this sample, you can enter *VALUE1, *VALUE2, or nothing. If you enter nothing, *VALUE1 is assumed.

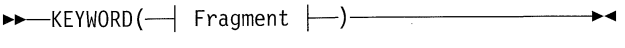


- **Repeated values** can be specified for some parameters. In this sample, the footnote indicates that you can enter up to 10 repetitions of the values.

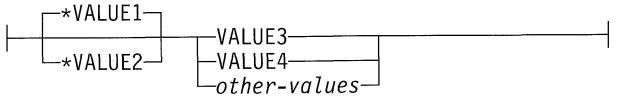


Note:
1 A maximum of 10 repetitions can be specified.

- **Fragments** allow you to label entire blocks of parameters or values and in another section of the syntax diagram.



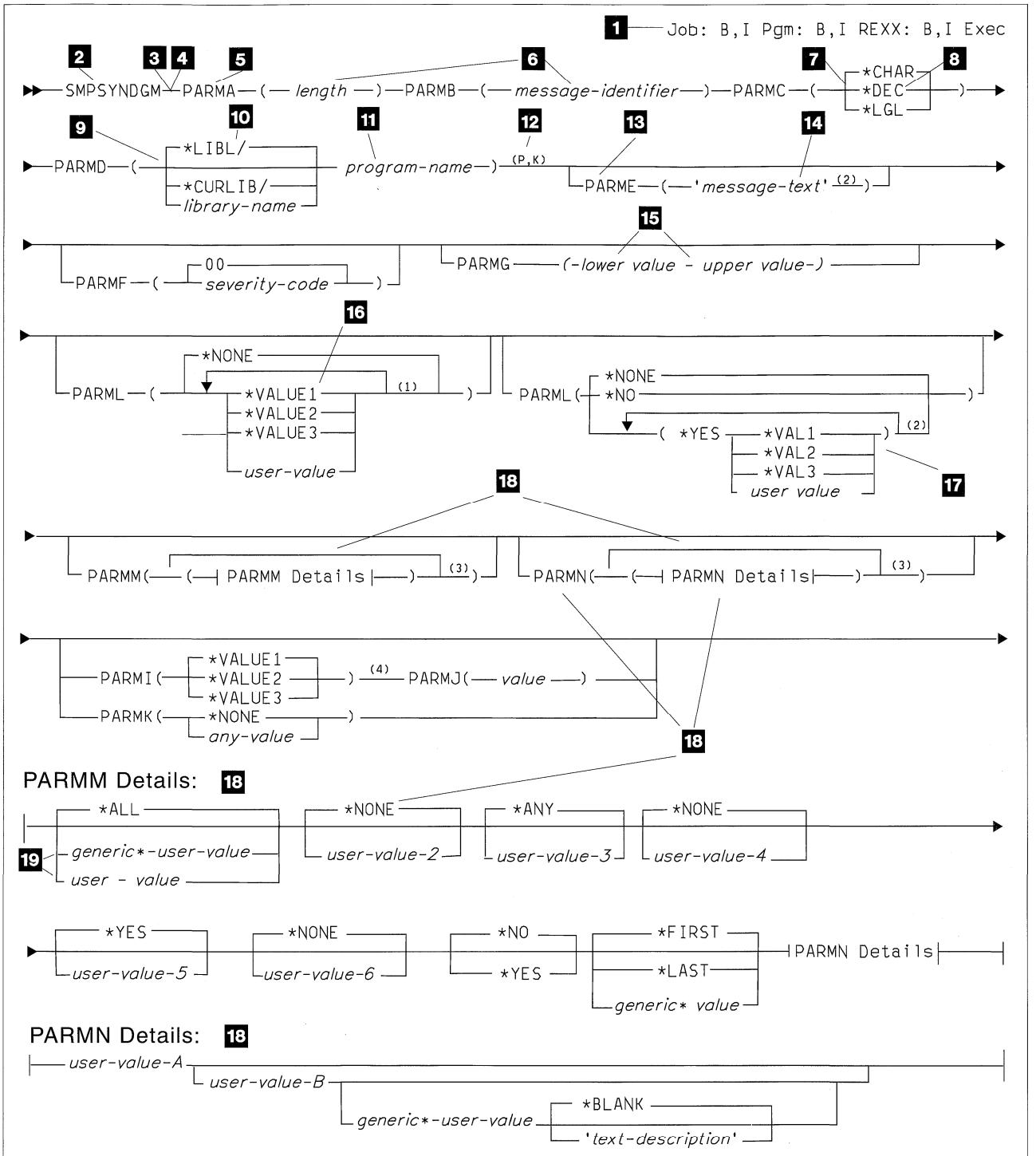
Fragment:



How Syntax Diagrams Show Command Structure

In addition to showing simple command structures such as parameter keywords and values, syntax diagrams also show more complicated command structures such as mutually exclusive parameters. The following section describes in more detail how the syntax diagrams show these command structures.

Sample Syntax Diagram: A sample syntax diagram is illustrated on the following page. It shows the parameter syntax of a fictitious command named Sample Syntax Diagram (SMPSYNDGM). This sample syntax diagram shows several parameter structures and how those structures are represented on the syntax diagram. In addition, the sample syntax diagram contains additional information about the command, its parameters, and its values (such as where the command can be used, see **1**). The rules on how to interpret this diagram are in the section that follows this syntax diagram, see "Syntax Diagram Rules."



Notes:

1 A maximum of 3 repetitions.
2 A maximum of 4 repetitions.
3 A maximum of 50 repetitions.
P All parameters preceding this point can be specified positionally.
K All parameters preceding this point are key parameters.
4 PARMJ is required only when PARMI(*VALUE2) is specified.

RV2F059-3

Syntax Diagram Rules: The syntax diagrams for the CL commands are interpreted according to the following rules.

1 *Entry Codes (Batch and Interactive)*

The box insert in the upper right corner of each syntax diagram contains the entry codes that specify the environment in which the command can be entered. The codes indicate whether the command can be:

- Used in a job (outside a compiled program; Job:B and/or I). When used in this manner, the command is considered a separate entity inside the job, and is run by itself as a separate function in what is called the interpretive mode. That is, commands in batch mode and/or interactive jobs that are not in compiled programs are interpreted and run one at a time, one after the other. The function of one interpreted command in the job is completed before the next command is interpreted.
- Used in a compiled program (Pgm:B and/or I). In this case, the command is part of the program; the command is in compiled form with the rest of the program, and running the command depends upon when the program is called and upon the program's logic preceding the command. That is, a compiled command cannot be run unless the program it is in is run.
- Used in a REXX procedure (REXX:B and/or I). In this case, the command is part of the REXX procedure. Running the command depends upon when the REXX procedure is started with the Start REXX Procedure (STRREXPRC) command. That is, the command cannot be run unless the procedure containing it is started.

The explanations of the combinations of entry codes are shown in the following chart:

Table 3. Entry Codes (Batch and Interactive)

Code	Representing	Meaning
Job:B	Batch job	Valid in batch input stream, external to compiled CL programs.
Job:I	Interactive job	Valid for interactive entry, external to compiled CL programs.
Job:B,I	Batch and interactive jobs	Valid for batch and interactive entry, external to compiled CL programs.
Pgm:B	Program, batch	Valid in a compiled CL program that is called from batch entry.
Pgm:I	Program, interactive	Valid in a compiled CL program that is called from interactive entry.
Pgm:B,I	Program, batch and interactive	Valid in a compiled CL program that is called from batch or interactive entry.
REXX:B	REXX, batch	Valid in a REXX procedure that is called from batch entry.
REXX:I	REXX, interactive	Valid in a REXX procedure that is called from interactive entry.
REXX:B,I	REXX, batch and interactive	Valid in a REXX procedure that is called from batch or interactive entry.
Exec	QCAPCMD and QCMDEXC processing	Valid as a parameter on the CALL command; can be passed as a character string to the system programs QCAPCMD and QCMDEXC.

2 *Command Name*

The command name appears first in the diagram. In the sample syntax diagram, the name of the fictitious command is SMPSYNDGM.

3 *Command Labels*

The labels that may precede all coded commands are optional. Therefore, labels are not shown in the syntax diagrams. If used, a label must be followed by a colon (:) immediately after the last character in the label name.

4 *Parameter Order*

All parameters of the command are shown in the correct positional sequence. The order goes left to right on each line and continues on the following line. To show the positional order of the parameters in the sample diagram, the representative parameter keyword names are named in alphabetic order, such as PARMA and PARMB. They are named in the same order in which they would have to be coded positionally if this command actually existed.

Note: In the few cases where dependent parameter relationships are shown (see rule **19**), the positional parameter order may not be readily apparent. The parameter order may be specified in a note in the diagram, or it can be easily determined from the text where all parameters in the command are described in positional order.

When coding parameters positionally, you must enter them in the order shown in the diagram. If you choose not to code a parameter and another positional parameter is coded after it, then you must enter *N to represent the uncoded parameter in the positional sequence.

Parentheses must be coded in each parameter that either has the keyword coded with its value (see rule **5**) or has multiple values in one parameter (see rule **17**).

5 *Parameter Keywords*

For each parameter, the keyword is always shown first, followed by the parameter values. Parameter keywords use

uppercase letters; if you code them in lowercase, they will be changed to uppercase.

When a parameter is coded in keyword form, its associated values must be enclosed in parentheses as shown below:

This parameter is codes as follows:

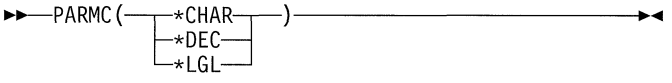
```
PARMA(15)
```

6 *User-Defined Values*

User-defined values are shown with lowercase characters that describe the kind of value coded by the user. If more than one word is used to describe a single value, the words are connected by hyphens:

7 *Choice of Values*

For a parameter having a choice of values, only one of which can be specified (typed after the parameter keyword and enclosed in parentheses), the choices of values are shown on different *branch* lines that follow the parameter keyword (which is on the *base* line), as follows:

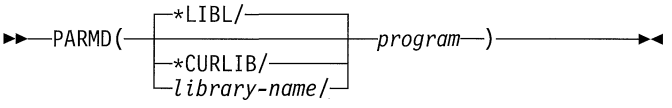


8 *Predefined Values*

Predefined values are shown exactly as they must be coded. Predefined values usually begin with an * followed by all uppercase letters. *CHAR and *DEC are examples of predefined values. Predefined values can also begin with the letter Q. For example, an option for the SRCFILE parameter on the CRTCLPGM command is QCLSRC.

9 *Optional Values*

Optional values are shown by a blank line as shown below:



The blank line indicates that a value from the first group (which includes *LIBL, *CURLIB, and *library-name*) does not have to be entered as shown below:

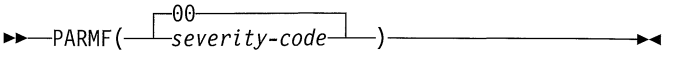
```
PARMD(CLPGM)
```

If the blank line were missing, you would have to code the parameter as shown below:

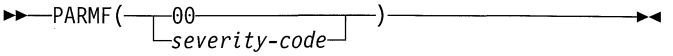
```
PARMD(*LIBL/CLPGM)   or
PARMD(*CURLIB/CLPGM) or
PARMD(MYLIB/CLPGM)
```

10 *Default Values*

If a parameter has a default value, it is indicated by its position above the base line. The value *LIBL (see **9**), is a default library qualifier on the PARMD parameter, and 00 is the default value on the PARMF parameter as shown below:



If 00 were not a default value, it would be shown as follows:



11 *Qualified Object Names*

Qualified object names include the optional library qualifier followed by the object name. If the qualifier is not specified, the default (*LIBL in this case) is used. Usually, *LIBL is the default value for a qualified object name; it means the system is to search the library list associated with the job to find the object.

For PARMD, the syntax diagram shows that a name can be specified in qualified or unqualified form. See **9** for coded examples.

12 *Positional Coding Limit and Key Parameter Limit*

The point to which the command's parameters can be coded positionally is indicated by a P on the base line.

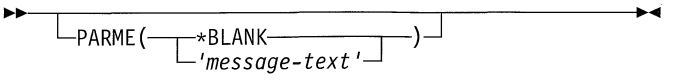
All parameters preceding the P can be coded without keywords (that is, positionally); all parameters after the P must be coded with their keywords. If you attempt to code a parameter positionally beyond this point, a syntax error will occur. When all parameters of a command can be coded positionally or when parameters do not exist for the command, a P does not appear on the syntax diagram.

A command can also have *key parameters*, which are the only significant parameters needed by the system to determine the actual values for other parameters on the command. Key parameters are the only parameters shown on the display when a user prompts for the command. After values are entered for the key parameters, the remaining parameters are shown with actual values instead of the default values (such as *SAME or *PRV).

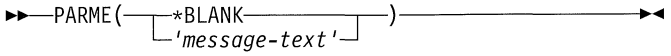
Key parameters include all parameters preceding the key parameter limit, which is designated on the syntax diagram by a K. If no K is shown, there are *no* key parameters on the command.

13 *Optional Parameters and Required Parameters*

Optional parameters do not have to be coded. Optional parameters are shown below the base line as follows:



If this parameter were required, it would appear on the base line as follows:



In most cases, required parameters precede optional parameters on the syntax diagram.

14 *Quoted Values*

User-defined variables that must be enclosed in apostrophes are shown enclosed with apostrophes in the diagram. Apostrophes are shown where one or more special characters are normally expected to occur among alphanumeric characters.

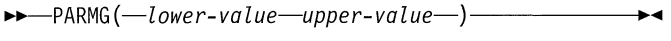
The value specified for PARME requires apostrophes if more than one word is entered or if special characters are used. Blanks, such as between words, are not allowed in an unquoted character string. PARMJ requires apostrophes if a character other than an alphanumeric character is specified.

```
PARME('This is a quoted string')
PARME('10-24-78')
PARME(102478)
```

The first and second values require apostrophes because they have either blanks (spaces) or special characters (-). The third value is a date with no separator characters; therefore, apostrophes are not required.

15 *Ordered List of Values*

Ordered lists of values (that is, values that must be specified in a particular order) will be shown as follows:



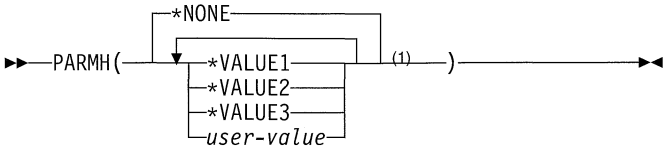
Note: An ordered list of values can contain predefined values, see **17** .

Both PARMG and PARMI show a list of two values that must be coded in the order shown when the parameter is coded. Parentheses are required around the list of values even if no keyword is used. PARMG and PARMI could be coded as:

```
PARMG(0  16)        PARMI(1  9999)
```

16 *Unordered List of Values (with repetition)*

A parameter that can have several values specified in no particular order (an unordered list of like values) is shown as follows:



Note:
1 A maximum of 3 repetitions.

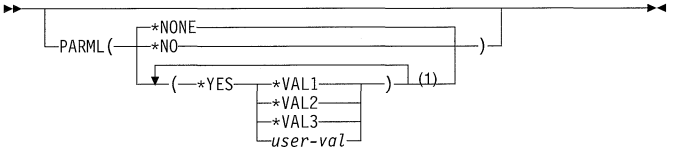
The PARML parameter can be specified as one, two, or three of the values shown. As the syntax note shows, any combination of three values selected and a user-defined value can be specified in any order. The user-defined value could be any value allowed for that parameter. Any of the following could be coded:

```
PARML(*VALUE1  *VALUE3)
PARML(*VALUE3 *VALUE2 16)
PARML(16 3 12)
```

If PARML is not specified, the single value *NONE is the default used by the system. Note that if *NONE is specified, none of the other values can be specified.

17 *Ordered List of Values (with repetition)*

A parameter that can have more than one ordered list of values (a series of ordered lists) of values is shown below.



Note:
1 A maximum of 4 repetitions.

PARML can be specified as a single value (*NONE or *NO) or as a list of values, but not as both. When PARML is coded as a list of values, the values must be specified in the order shown, and both elements in the list must be specified (in other words, *YES must always precede either a predefined or user-defined value). For example, PARML can be coded as follows (notice that two sets of parentheses are required for the elements):

```
PARML((*YES *VALUE1))
PARML(*NONE)
PARML((*YES 50))
```

However, PARML *cannot* be coded as follows:

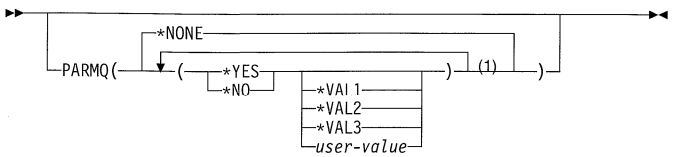
```
PARML((*NONE *VALUE1))
PARML((*VALUE1 *YES))
PARML(*VALUE1)
PARML(*YES)
```

If repetition is used to specify more than one list of values for PARML, each list must be enclosed in parentheses. For example, three repetitions (the maximum for PARML) of lists of values could be coded as follows:

```
PARML((*YES *VALUE2) (*YES *VALUE1) (*YES 43))
```

Notice that a particular order is required for the elements in each list, but a particular order is not required for repeated lists. The value *YES must always precede the other values within the list, but (*YES *VALUE1) does not have to be the first list specified for PARML.

For some commands, the second element is not required as shown below:



Note:
[1] A maximum of 3 repetitions.

Notice that in this example, that *VAL1 can be specified by itself, and the second element does not have to be specified. However, if the first element is coded by itself, the extra set of parentheses are required as shown below:

```
PARMQ((*VAL1) (*VAL1 *VAL3) (*VAL2))
```

18 *Ordered Lists of Values as Fragments*

In this manual, fragments are used to show and reference long lists of values, as shown on **18** in the sample syntax diagram. If there are fewer than five elements, the structure shown for PARMN is used (see PARMN details on the sample syntax diagram). The structure of PARMN correctly shows that the elements must be coded positionally. You cannot code *BLANK without coding the first three values or including the place holder, *N. The following could be entered:

```
PARMN(A)
PARMN(A B)
PARMN(A B GENVAL*)
PARMN(A *N GENVAL* *BLANK)
```

Note: You can use *N as a place holder.

However, the following would yield incorrect results because the elements are not coded positionally.

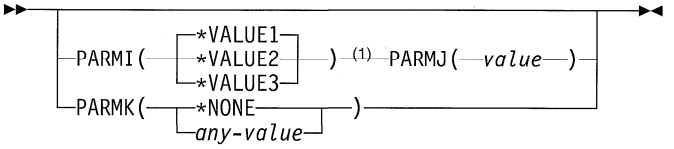
```
PARMN(GENVAL* *BLANK)
PARMN(*BLANK)
PARMN(B 'User Object')
```

If a parameter contains five or more elements, the structure shown for PARMM is used (see PARMM details in the sample syntax diagram) because using the structure shown for PARMN for large numbers of elements makes the syntax diagram too complex. These elements are all shown as optional. However, they, like the elements in PARMN, must be coded positionally.

Fragments can contain fragments as shown in PARMM details on the sample syntax diagram. PARMM details contains all of the PARMN details all of which must be specified positionally.

19 *Dependent Parameter Relationships*

Some parameters or values have dependent relationships with other parameters or values. Parameters that dependent on values are indicated by a footnote. Parameters that are mutually exclusive are shown in the example below:



Note:
[1] PARMJ is required only when PARMI(*VALUE2) is specified.

PARMI and PARMK are mutually exclusive parameters; PARMI and PARMK cannot be coded on the same command. As the note shows, PARMJ is a dependent parameter which is valid only when *VALUE2 is specified for the PARMI parameter. Note that PARMJ cannot be specified if PARMK is specified. Also note that all of these parameters are optional because they appear below the base line.

Part 2. Command Definition

Chapter 5. Command Definition Statements

Command Definition Statements

The AS/400 control program lets users define a command that calls a program to perform some function. Users can define commands by using a command definition statement. The defined command can include the following:

- Keyword notation parameters for passing data to programs
- Default values for omitted parameters
- Parameter validity checking so the program performing the function will have correct input
- Prompt text for prompting interactive users

Creating User-Defined Commands

Users can define a command by entering command definition statements into a source file and running a Create Command (CRTCMD) command using the source file as input. One and only one Command (CMD) statement must be somewhere in the source file. A Parameter (PARM) statement must be provided for each parameter that appears on the command being created. If any special keyword relationships need checking, the Dependent (DEP) statement is used to define the relationships. The DEP statement can only refer to parameters that have been previously defined. These statements can appear in any order. The *CL Programmer's Guide* contains a complete description of how to use these statements to define a command.

Only one command can be defined in each source member in the source file. The CRTCMD command is run to create the command definition object from the command definition statements in one source file member. Other users can then be authorized to use the new command by the Grant Object Authority (GRTOBJAUT) command or the Edit Object Authority (EDTOBJAUT) command.

Command Definition Statement Descriptions

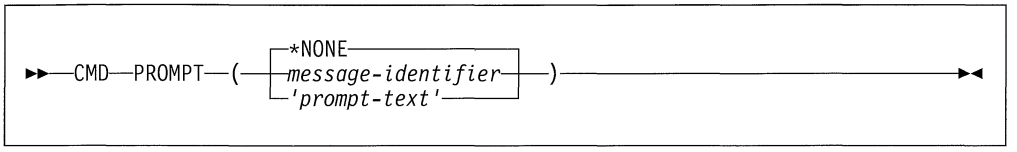
The **command definition statement** of each command contains one or more of the following **command statements**:

- CMD (Command)
- DEP (Dependent)
- ELEM (Element)
- PARM (Parameter)
- PMTCTL (Prompt Control)
- QUAL (Qualifier)

These command statements are listed (in alphabetical order) and described in this section.

CMD (Command) Statement

The Command (CMD) statement specifies the prompt text for the command being created. The prompt text is displayed at a work station when a user requests prompting while entering the command that is being defined. The CMD statement can be anywhere in the source file referred to by the Create Command (CRTCMD) command; one and only one CMD statement must be used in the source file, even if no prompt text is specified for the created command.



PROMPT Parameter

Specifies the prompt text, if any, that is included in the heading (title) of the prompt display for the command being defined. The prompt text further describes the name of the command. For example, in the CRTLIB prompt heading, "Create Library (CRTLIB) Prompt," the words *Create Library* would be specified as the prompt text in this PROMPT parameter.

Note: Prompt text for each of the *parameters* in this command can be specified in the PROMPT parameters of the PARM, ELEM, and QUAL command definition statements. They specify the prompt text for the parameters, just as the PROMPT parameter in the CMD statement specifies the prompt text for the command (in the heading).

***NONE:** No prompt text is included in the displayed heading of the prompt when the command is being prompted.

message-identifier: Specify the message identifier that specifies the message, containing no more than 30 characters, for the prompt text that is shown when the command is being prompted. If a message having the specified identifier cannot be found in the message file specified in the PMTFILE parameter of the Create Command (CRTCMD) command, the message identifier itself is used as the prompt text.

'prompt-text': Specify the prompt text that is displayed during the command prompting. It must be a character string of no more than 30 characters, enclosed in apostrophes.

Note: Variables cannot be coded for this parameter.

Example

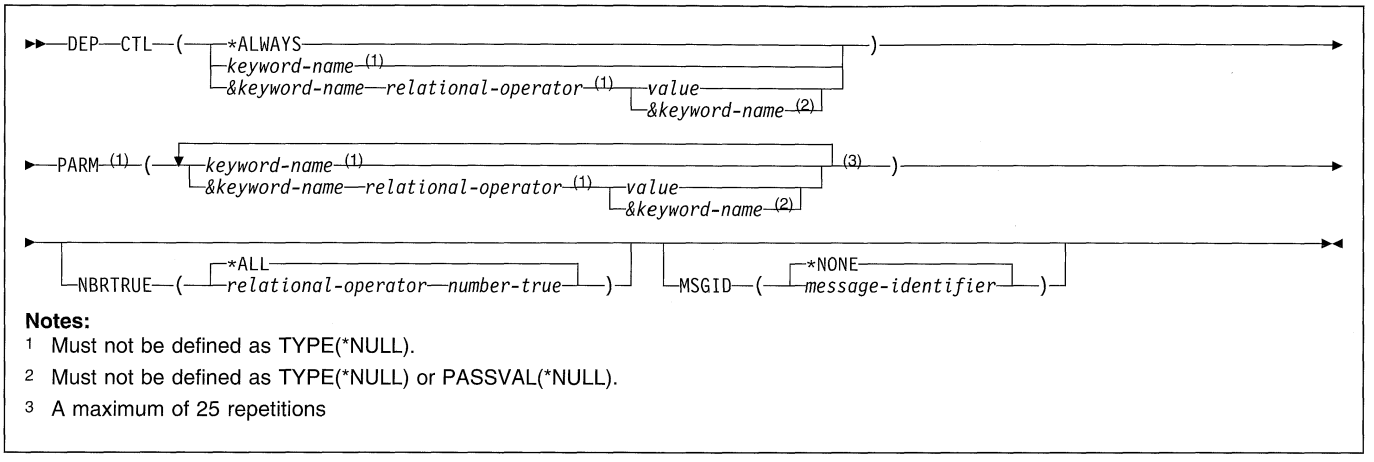
```
CMD  PROMPT(UCD0001)
```

This statement describes a command that is prompted with additional text in the display heading; the prompt text comes from the message identified by UCD0001.

DEP (Dependent) Statement

The Dependent (DEP) statement defines a required relationship between parameters and parameter values that must be checked. This relationship can refer to either the specific value of a parameter or parameters or to the required presence of parameters.

If a parameter has a default value and the parameter is not specified, the checking differs depending on whether the DEP statement is performing a specification check or a relational check. If a *specification* check is made on an unspecified parameter (checking for the presence of a value for that parameter), the system assumes that no value was specified, and the default value is *not* used. If a *relational* check is made on an unspecified parameter, the default value is used as the parameter value.



Notes:
1 Must not be defined as TYPE(*NULL).
2 Must not be defined as TYPE(*NULL) or PASSVAL(*NULL).
3 A maximum of 25 repetitions

CTL Parameter

Specifies the controlling conditions that must be true before the parameter dependencies defined in the PARM statement must be true. The first keyword specified identifies the controlling parameter. The controlling condition can be specified either by a keyword name only or by a keyword name and a test relationship that determines whether the controlling condition requires the presence of the parameters it is dependent upon. The relationship between the controlling parameter and a specified value can be tested to determine whether the condition specified is met. If it is, the parameters that the controlling parameter is dependent upon must meet the requirements specified in the PARM and NBRTRUE statements.

***ALWAYS:** The parameter dependency is always checked, regardless of the form of the command.

keyword-name: Specify the keyword name of the parameter for which a value must be specified to control dependency. The keyword name is the name of the parameter that is specified by the KWD parameter on the PARM statement defining it. If the keyword is specified, the parameter dependency is checked. The keyword must not be defined as TYPE(*NULL).

&keyword-name relational-operator value: Specify the keyword name of the controlling parameter followed by a relational operator (such as *LE or *EQ) and a value to be tested. If the tested condition is met, the parameters that the controlling parameter is dependent upon must meet the requirements specified in the PARM statement. The value must be no longer than 32 characters. The keyword must not be defined as TYPE(*NULL).

If the value being tested against has been specified as a special value or single value (SPCVAL or SNGVAL parameters of the PARM statement), the to-value must be used rather than the from-value.

The keyword name must be preceded by an ampersand (&) to indicate that the value of the keyword is tested if the relational operator and value are specified; the ampersand must not be used if the relational operator and value are not specified.

(&keyword-name relational-operator &keyword-name): Specify the keyword name of the controlling parameter followed by a relational operator (such as *EQ) and the keyword name of another parameter whose value is compared with the value of the controlling parameter. The keywords must not be defined as TYPE(*NULL) or with PASSVAL(*NULL) specified.

PARM Parameter

Specifies the parameter dependencies that must be tested if the controlling conditions defined by the CTL parameter are true. The dependencies can be one or all of the following:

- The names of one or more parameters that will be tested to determine if they are present.
- One or more test relationships of keyword values to the values of other keywords or constants. A maximum of 25 parameters can be specified for the

PARM parameter. Keywords specified in this parameter must not be defined as TYPE(*NULL).

keyword-name: Specify the keyword name of each parameter that must have a value specified for it.

&keyword-name relational-operator value: Specify the keyword name of each parameter followed by a relational operator and a value to be tested. An ampersand must precede the keyword name to indicate that the value of the keyword is tested. The value specification must be no longer than 32 characters.

If the value being used for testing comparison has been specified as a special value or single value (SPCVAL or SNGVAL parameters of the PARM statement), the to-value must be used rather than the from-value.

&keyword-name relational-operator &keyword-name: Specify the keyword name of one parameter followed by a relational operator and the keyword name of another parameter whose value is compared with the value of the first parameter. A keyword defined with PASSVAL(*NULL) cannot be specified.

NBRTRUE Parameter

Specifies the number of parameter dependencies defined in the associated PARM statement that must be true to satisfy the control condition.

***ALL:** All the parameter dependencies specified in the PARM statement must be true to satisfy the control condition.

relational-operator number-true: Specify a relational operator and a number that specifies the number of parameter dependencies that must be true to satisfy the specified relation. For example, if (*EQ 2) is specified for NBRTRUE, then two, and only two, of the parameter dependencies must be true for the PARM statement.

Note: Variables cannot be coded for this parameter.

MSGID Parameter

Specifies the message identifier of an error message in a message file that is sent to the user if all the specified parameter dependencies have not been satisfied.

***NONE:** No special message is sent. Instead, a message generated by the command analyzer is sent to the user.

message-identifier: Specify the message identifier of the error message sent to the user if all the dependencies for this statement are not met. Messages whose identifiers begin with the 3-character prefixes *CPF* or *CPD* are retrieved from the IBM-supplied message file QCPFMSG. All other messages specified here are retrieved from the message file identified by the MSGF parameter on the CRTCMD command which is used to create the command being defined with these dependencies.

Note: Variables cannot be coded for this parameter.

Examples

```
DEP CTL(&TYPE *EQ LIST) PARM(ELEMLIST)
```

If TYPE(LIST) is specified, the ELEMLIST parameter must be specified.

```
DEP CTL(FILE) PARM(VOL LABEL) NBRTRUE(*EQ 2)
```

If the FILE parameter is specified, both the VOL and LABEL parameters must be specified.

```
DEP CTL(GLOOP) PARM(J1 D J2) NBRTRUE(*EQ 1)
```

If the GLOOP parameter is specified, a value must be specified for one (and only one) of the J1, D, and J2 parameters.

```
DEP CTL(&LIB *EQ MYLIB)
PARM((&PASSWORD *eq XYZ5)
    (&USRPRF *EQ B0BJ))
NBRTRUE(*GE 1) MSGID(MSG001)
```

If the LIB parameter equals MYLIB, then the PASSWORD parameter must equal XYZ5, or the USRPRF parameter must equal BOBJ. Otherwise, message MSG001 will be sent to the user.

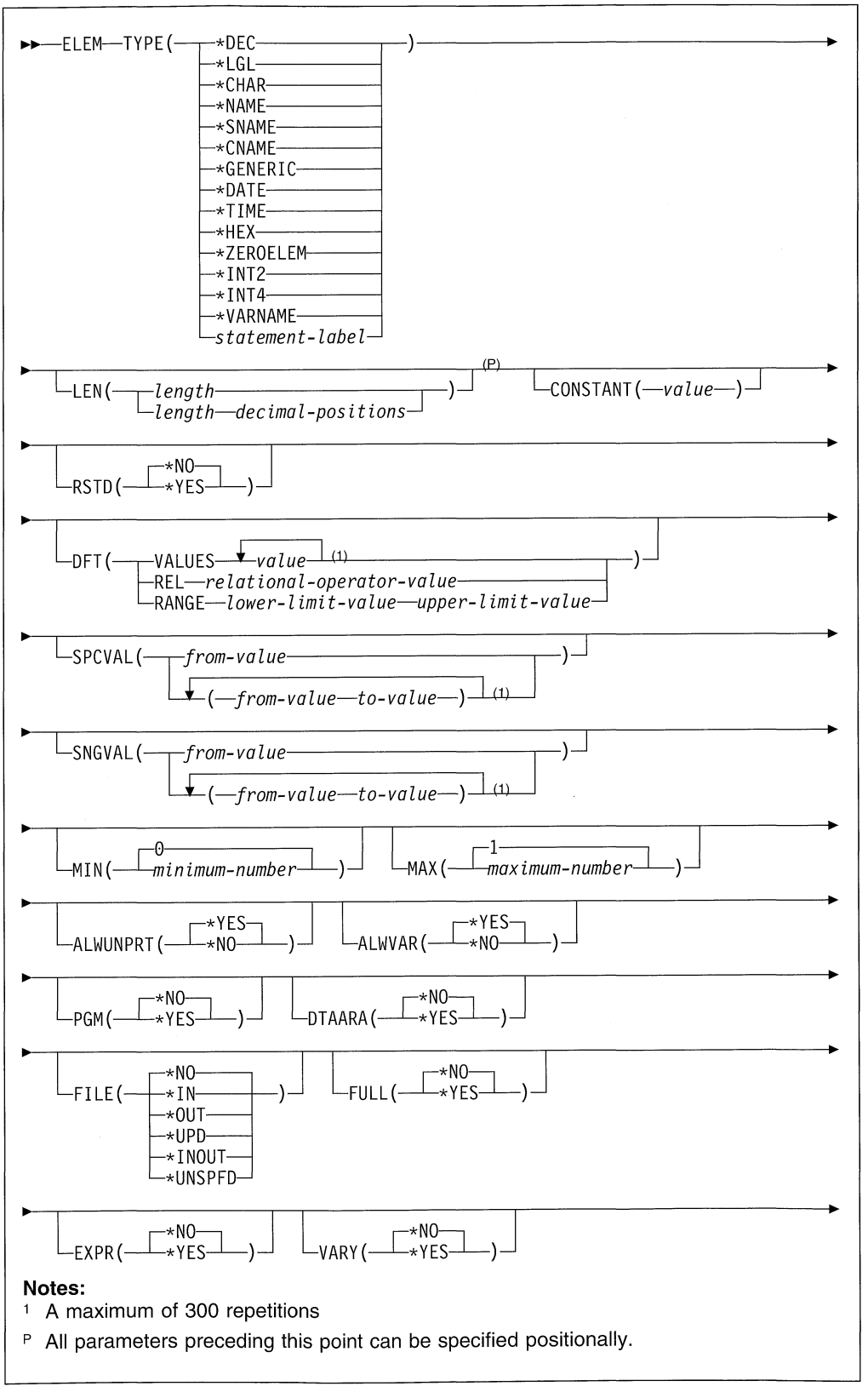
ELEM (Element) Statement

Element (ELEM) statements are used to define the elements of a mixed list (list elements) parameter on a command. A list parameter is a parameter that accepts multiple values that are passed together as consecutive values pointed to by a single keyword. The values are preceded by a 2-byte binary value that indicates the number of elements defined for the parameter. CL programs do not support binary values in variables.

A list element is the value that represents one value among a group of values organized in a specific order in a list. If all of the list elements are not of the same type, one ELEM statement must be used for each element that appears in the list being defined. If all the elements are of the same type (a simple list), individual ELEM statements are not required. For a simple list, specify the number of elements in the list on the MAX parameter of the PARM statement.

The order in which the ELEM statements are entered into the source file determines the position of the elements in the list. The first ELEM statement (for the first list element) must have a statement label that matches TYPE (statement-label) on the PARM or ELEM statements for the same list. The remaining ELEM statements in the list must be unlabeled. Lists of elements having different values can be nested to the depth of three levels including the highest level. A maximum of 300 elements can be included in one list.

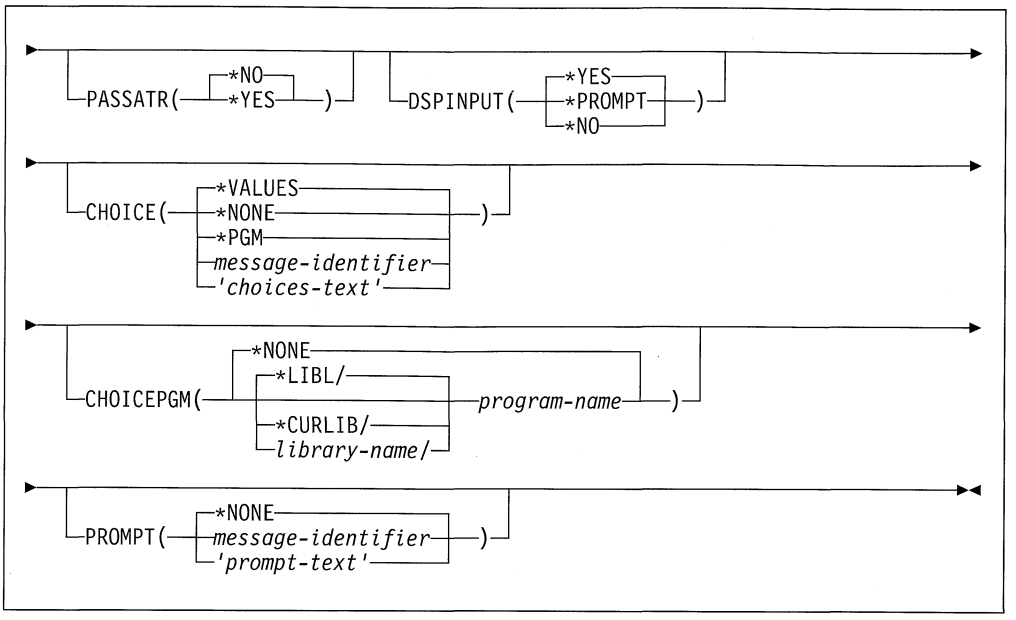
Note: The ELEM statement contains certain parameters and predefined values that can be used only when an IBM-supplied command processing program (CPP) is called by the command being defined. Because there are limitations in some high-level languages, these values may not be useful in the definition statements of user-defined commands. If the entire parameter is for IBM-supplied commands only, these parameters and values are identified by placing the phrase, "(*For IBM-supplied commands)*" immediately following the parameter keyword or the predefined value to which it applies.



Notes:

1 A maximum of 300 repetitions

P All parameters preceding this point can be specified positionally.



TYPE Parameter

Specifies the type of list element being defined. The element can be an integer, a decimal or logical value, or a quoted or unquoted character string that can be a name, label, date, or time. Enter one of the following values to specify the type of element.

Note: For more information on names (below), see "Rules for Specifying Names" in chapter 2.

***NAME:** The list element is a character string that represents a basic name. The maximum length of the name is 256 characters; the first character must A-Z, $, @, or #. The remaining characters can be the same as the first character but can also include the numbers 0 through 9, underscores (_), and periods (.). The name can also be a string of characters starting and ending with double quotation marks ("). If a special value is used (as in *LIBL or *NONE), it must be specified in the SPCVAL parameter.

***SNAME:** The list element is a character string that represents a simple name. The maximum length of the name is 256 characters; the first character must be A-Z, $, @, or #. The remaining characters can be the same as the first but may also include the numbers 0 through 9 and underscores (_). Periods (.) are not allowed. If a special value is used (such as *LIBL or *NONE), it must be specified in the SPCVAL parameter.

***CNAME:** The list element is a character string that represents a communications name. The maximum length of the name is 256 characters; the first character must be A-Z, $, @, or #. The remaining characters can be the same as the first character but may also include the numbers 0 through 9. Periods (.) and underscores (_) are not allowed. If a special value is used (such as *LIBL or *NONE), it must be specified in the SPCVAL parameter.

***GENERIC:** The list element is a character string that represents a generic name. A generic name contains a maximum of 255 characters followed by an asterisk (*) or 256 characters without an asterisk and must conform to the rules for generic names. The name identifies a group of objects whose names all begin with the characters preceding the *. If an * is not included, the system assumes that the generic name is a complete object name.

***DATE:** The list element is a character string that represents a date. When it is passed to the CPP, it is always passed in the format *Cyymmdd*, where C = century, yy = year, mm = month, and dd = day. The century digit is set to 0 (zero) if yy is equal to or greater than 40 (1940 through 1999); it is set to 1 (one) if yy is less than 40 (2000 through 2039). When a date value is specified in this ELEM statement, it must be specified without quotation marks in the format *mmddyy*. If the user enters a date when the command is run, it must be entered in the format specified by the system value QDATFMT. The system value QDATSEP specifies the optional separator character that must be used when the date is entered. If the separator character is used, the date must be enclosed in apostrophes.

***TIME:** The list element is a character string that represents a time. It is in the format *hh:mm:ss*, where hh = hours, mm = minutes, and ss = seconds. It is passed to the CPP in a 6-byte character string as *hhmmss*. Values specified in this statement must be in the format *hhmmss*.

***CHAR:** The list element is a character string that optionally can be enclosed in apostrophes. If the character string contains any special characters (not including an asterisk (*)), it must be enclosed in apostrophes. The maximum length of the character string is 3000 characters.

***DEC:** The list element is a packed decimal number.

***LGL:** The list element is a logical value, either a one ('1') or a zero ('0').

***HEX:** The list element value is in hexadecimal form. The specified characters must be 0 through F. They are converted to hexadecimal (EBCDIC) characters (2 hex digits per byte), right justified, and padded with zeros. If the value is enclosed in apostrophes, an even number of digits is required. If the value is not enclosed in apostrophes, the number of digits may be odd or even.

***ZEROELEM:** The list element is always considered to be a list of zero elements for which no value can be specified in the command. It is used to prevent a value from being entered as an element in a list even though the CPP expects one. An element for which *ZEROELEM is specified is not prompted, although the other elements in the parameter are prompted and are passed to the CPP as a list.

***INT2:** The list element is an integer that is passed as a 2-byte signed binary number. CL programs do not support using binary values in variables.

***INT4:** The list element is an integer that is passed as a 4-byte signed binary number. CL programs do not support using binary values in variables.

***VARNAME:** (For IBM-supplied commands) The list element is a variable name that is passed as a character string. The name can contain a maximum of 11 characters, including the initial ampersand (&).

statement-label: Specify a qualified name or a mixed list of values. The statement label specified here by the TYPE parameter is the statement label that identifies the first of a series of QUAL or ELEM statements that further describe the qualified name or the mixed list being defined. The label must be the same as the label specified by TYPE(statement-label) on the PARM statement for this list.

LEN Parameter

Specifies the length of the list element value that is passed to the CPP. If TYPE was specified as *INT2, *INT4, *DATE, *TIME, *ZEROELEM, or statement-label, LEN is not allowed. If TYPE(*DEC) was specified, the decimal length is specified in the form (n1 n2), where n1 specifies the total number of digits in the value (including the decimal portion), and n2 specifies the number of allowable decimal digits to the right of the decimal point. The value for n2 is optional, and zero is assumed if n2 is not entered. If TYPE is other than *DEC, the decimal portion (n2) must be omitted and only the number of characters must be specified.

Table 4. Lengths of Data Types That Can Be Coded

Data Type	Default Length	Maximum Length[1]
*DEC	(15 5)	(24 9)
*LGL	1	1

Table 4. Lengths of Data Types That Can Be Coded

Data Type	Default Length	Maximum Length[1]
*CHAR	32	3000
*NAME	10	256
*SNAME	10	256
*CNAME	10	256
*GENERIC	10	256
*HEX	1	256
*VARNAME	11	11

1 The maximum length shown here is the maximum length allowed by the Command Definition. The high-level language used as the CPP for the command may have different maximum lengths for these data types (for example, *DEC values in CL programs have a maximum length of (15 9)).

Whereas the maximum shown here is the maximum for the values used in the command when it is run, the maximum length of character constants specified in the command definition is limited to 32 characters. This restriction applies to the following parameters: CONSTANT, DFT, VALUES, REL, RANGE, SPCVAL, and SNGVAL.

For data types whose length cannot be coded, the following are the maximum lengths and the lengths passed.

Table 5. Lengths of Data Types That Cannot Be Coded

Data Type	Maximum Length	Length Passed
*DATE[1]	8	7
*TIME[2]	8	6
*ZEROELEM	0	0
*INT2[3]	6	2
*INT4[4]	11	4
statement-label[5]	–	–

1 If a date is specified, the value is passed as 7 characters.
2 If a time is specified, the value is passed as 6 characters.
3 The value must meet the following condition: $-2^{15} \leq$ value $\leq 2^{15}-1$. The value is passed as a 2-byte signed binary number.
4 The value must meet the following condition: $-2^{31} \leq$ value $\leq 2^{31}-1$. The value is passed as a 4-byte signed binary number.
5 The length of the data accepted and passed is defined by the ELEM or QUAL statement that the label identifies.

CONSTANT Parameter

Specifies that a value is passed to the CPP as a constant for the list element when the command being defined is processed. The value does not appear externally on the command. If specified, the value must satisfy the requirements specified by the TYPE, LEN, VALUES, REL, RANGE, SPCVAL, and FULL parameters. As noted in the LEN parameter chart, if a character constant is specified in this parameter, it can be no longer than 32 characters. CONSTANT is not valid with TYPE(*ZEROELEM), EXPR(*YES), or MAX(>1), or if DFT was coded for ELEM.

If a constant is specified for the element being defined, no prompt text can be specified for the PROMPT parameter of this ELEM statement. However, the other elements of the list parameter (of which this list element is a part) are still prompted, and their values (along with this constant value) are still passed to the CPP as a list.

Variables cannot be coded for this parameter.

RSTD Parameter

Specifies whether the value entered for the list element (specified in the ELEM statement) is restricted to only one of the values given in the VALUES, SPCVAL, or SNGVAL parameters, or whether the value can be any value that satisfies the requirements specified by the TYPE, LEN, REL, RANGE, SPCVAL, SNGVAL, and FULL parameters.

***NO:** The value entered for the list element defined by this ELEM statement can be anything that matches the requirements specified by the TYPE, LEN, REL, RANGE, SPCVAL, SNGVAL, and FULL parameters in this ELEM statement.

***YES:** The value entered for the list element in this ELEM statement is restricted to one of the values in the VALUES parameter, or to one of the from-values in the SPCVAL or SNGVAL parameter. *YES cannot be specified if TYPE(statement-label) or TYPE(*ZEROELEM) is specified.

DFT Parameter

Specifies the default value that is assigned to the list element. That is, the default value is used as the value of the list element if the user omits the parameter that represents this list element, or specifies *N for the element, while coding or entering the command. The default value must satisfy one of the following:

- It must match the element requirements specified by the TYPE, LEN, REL, RANGE, and FULL parameters.
- It must be one of the from-values in the SPCVAL or SNGVAL parameters.
- If the default is a character constant, it can have no more than 32 characters (as noted in the LEN parameter chart).
- If RSTD(*YES) is specified, it must be in the list of values in the VALUES parameter or in the list of from-values of the SPCVAL or SNGVAL parameters.
- If this ELEM statement itself defines a list, the default value must be specified in the SNGVAL parameter.

The DFT parameter is valid only if MIN is 0, which means the element defined by this ELEM statement for this list is optional. The DFT parameter is not valid if the CONSTANT parameter is specified. A default cannot be specified if TYPE(*ZEROELEM) is specified; in that case, an assumed default is passed.

If DFT is not specified, the default assumed is as follows, depending on the specified element type:

Assumed Default	Element Types
0	*DEC *INT2 *INT4
	*ZEROELEM
'0'	*LGL
zeros	*DATE *TIME *HEX
blanks	*CHAR *NAME *SNAME
	*CNAME *GENERIC
	*VARNAME

An *assumed* default value is not displayed by the command prompt; a blank input field is shown instead. If a default is specified in the DFT parameter, it is displayed by the prompt exactly as specified.

value: Specify the default value that meets the specified requirements or that is one of the values specified in the SPCVAL, SNGVAL, or VALUES parameters.

Variables cannot be coded for this value.

VALUES Parameter

Specifies a list of up to 300 constants (fixed values) from which one constant can be specified as the value of the list element. The VALUES parameter is valid only if all of the following are true: RSTD(*YES) is specified, both RANGE and REL are *not* specified, and each constant matches the attributes specified by the TYPE, LEN, and FULL parameters in this ELEM statement. As noted in the LEN parameter chart, character constants specified in this parameter can be no longer than 32 characters. Enter the constants (not more than 300) that can be specified as the value of the list element. The VALUES parameter is not valid for TYPE(statement-label) or for TYPE(*ZEROELEM).

If this ELEM statement is defining the first element in a list, the value specified for this parameter cannot be the same as the value specified in the SNGVAL parameter on either the PARM or ELEM statement that points to this ELEM statement.

REL Parameter

Specifies the relationship between the list element value and the value of another parameter or constant. The value associated with the referenced keyword, not the user-specified value, is passed to the CPP.

To specify the relationship, enter one of the following relational operators followed by a constant or the value of another parameter.

*LT	less than
*LE	less than or equal to
*EQ	equal to
*GE	greater than or equal to
*GT	greater than
*NL	not less than

*NE not equal to
*NG not greater than

The REL parameter is not valid if TYPE is specified as *LGL, *VARNAME, *ZEROELEM, or statement label; or if either RANGE or VALUES is specified.

If a character type is specified by TYPE(*CHAR), the EBCDIC value of the character string is used as an unsigned integer in the comparison. As noted in the LEN parameter chart, if a character constant is specified in this parameter, it can be no longer than 32 characters.

RANGE Parameter

Specifies the range (the limits) for the value of the list element. The list element value must be greater than or equal to the lower limit value specified, and it must be less than or equal to the upper limit value specified. The value tested is the value sent to the CPP, not the user-specified value. For example, 15 would be valid if RANGE was specified as (0 16).

For nonnumerical data types, such as *CHAR, the range of values and the data specified will be right-justified and padded with blanks. A numeric range should not be used to define an interval for nonnumerical data unless leading zeros are specified or the data is only 1 character in length.

The RANGE parameter is not valid if either REL or VALUES is specified, or if TYPE is specified as *LGL, *VARNAME, *ZEROELEM, or statement label. As noted in the LEN parameter chart, character constants specified in this parameter can be no longer than 32 characters.

SPCVAL Parameter

Specifies a list of up to 300 entries that define special values that can be entered for the element defined by this ELEM statement. Each entry specifies a character string (a from-value) that can be entered even though it may not meet all validity checking requirements. If the entered character string matches the from-value of one of the entries, and the to-value is specified, the string is replaced with the to-value and is then passed to the CPP without further checking. If the to-value is omitted, the from-value is passed to the CPP. The SPCVAL parameter is not valid for TYPE(statement-label) or for TYPE(*ZEROELEM).

If a to-value of *CURLIB is specified, the name of the current job library is passed to the CPP instead of the value *CURLIB. If the from-value is *CURLIB and no to-value is specified, or if the to-value is *CURLIB and it is enclosed in apostrophes, the value *CURLIB is passed to the CPP.

The from-value is a character string, but the to-value can be anything that is passable. However, for TYPE(*DATE), the to-value must be specified unquoted in the mmddyy format. If a CL variable is used for the from-value, its type must be *CHAR. If this ELEM statement is defining the first element in a list, the value specified for the from-value cannot be the same as the value specified in the SNGVAL parameter on either the PARM or ELEM statement that points to this ELEM statement.

The to-value must be no longer than LEN specifies. If TYPE is *DEC, *INT2, or *INT4, the type of the to-value must be the same. If TYPE is a character type (such as *CHAR, *LGL, or *DATE), the to-value must be a character string. As noted in the LEN parameter chart, character constants specified in this parameter can be no longer than 32 characters. If a to-value is not specified, the from-value must be passable.

Variables cannot be coded for this element.

SNGVAL Parameter

Specifies a list of up to 300 single values that can be specified for an element being defined as a statement label, or that is to have two or more list elements (defined by the MAX parameter) in its nested list. Any one of the single values can be used instead of a nested list of values or a qualified name that the element is defined to accept. Each entry specifies a character string (a from-value) that can be entered. If an entered character string matches the from-value of one of the entries and the to-value is specified, the data is replaced with the to-value and is then passed to the CPP without further checking. If the to-value is omitted, the from-value is passed to the CPP.

If this ELEM statement defines the first element in a list, the value specified for the from-value cannot be the same as the value specified in the SNGVAL parameter on either the PARM or ELEM statement that points to this ELEM statement.

The to-value (or the from-value, if the to-value is omitted) must be passable, as specified in the SPCVAL parameter. As noted in the LEN parameter chart, character constants specified in this parameter can be no longer than 32 characters. SNGVAL can be specified only if MAX is greater than one or TYPE is specified as a statement label; it is not valid for TYPE(*ZEROELEM). Each single value can only substitute for a list of values or a qualified name; it cannot be a list element or qualifier. It is passed as the first element of the list.

If a to-value of *CURLIB is specified, the name of the current job library is passed to the CPP instead of the value *CURLIB. If the from-value is *CURLIB and no to-value is specified, or if the to-value is *CURLIB and it is enclosed in apostrophes, the value *CURLIB is passed to the CPP.

Variables cannot be coded for this element.

MIN Parameter

Specifies the minimum number of values that must be entered for the list element being defined.

For an element that does not allow multiple like values, only zero (0) for optional and one (1) for required can be specified as the minimum number of values.

For an element that allows multiple like values because a value greater than one (1) is specified in the MAX parameter, zero (0) indicates that no values must be entered; therefore, it is an *optional* element. A value equal to or greater than one (1) indicates the minimum number of values that must be entered for the element. Therefore, it is a *required* element. The value specified for MIN cannot exceed the value specified for the MAX parameter.

The number specified tells how many list elements are required in another list. If MIN is not specified, zero (0) is assumed, meaning that the element is optional.

0: The list element is optional; it does not have to be entered.

minimum-number: Specify the minimum number of elements that must be specified in the nested list. If 1 is the assigned value, it specifies that at least one value is required for the element. If a number greater than one (1) is specified, the element contains a list that must have at least as many elements as the number specified.

MAX Parameter

Specifies, if this ELEM statement is defining a simple list element, the maximum number of elements that this list element can have in its nested list. If a value greater than 1 is specified, the element is capable of accepting multiple like values (that is, a simple nested list). All values entered for this element (at the time the command is run) must satisfy the validity checking requirements specified by the values specified in the other parameters on this ELEM statement.

Note: The values for a nested list are passed consecutively, preceded by a 2-byte binary value that indicates the number of values entered in the list element by the user. CL programs do not support the handling of binary values in variables.

1: The list element accepts only one value; there is no nested list.

maximum-number: Specify the maximum number of elements that the list element can accept. The specified maximum must be greater than or equal to the value specified in MIN, and less than or equal to 300. If the maximum is greater than one (1) and TYPE is not a statement label that identifies a QUAL statement or another ELEM statement, the parameter–which is also an element –is a simple list of like values (that is, each element in the list has the same requirements, such as type and length). If TYPE(statement-label) is specified and it points to the label of a QUAL statement or another ELEM statement, MAX should only be specified greater than 1 if a list of lists or a list of qualified names is to be accepted. A maximum greater than one (1) is not valid if the CONSTANT parameter is also specified.

ALWUNPRT Parameter

Specifies whether this ELEM statement should accept the hexadecimal characters X'FF' or those in the range of X'00' to X'3F'. This parameter is valid only for TYPE(*CHAR) or TYPE(*X).

***YES:** Allows any characters to be sent to the display or printer.

***NO:** Does not allow unprintable characters to be passed to the command processing program.

ALWVAR Parameter

Specifies whether to allow variable names for the element. If TYPE was specified as *VARNAME, *ZEROELEM, *NULL, or statement-label, ALWVAR(*NO) is not allowed.

***YES:** Allows the use of variable names for the element.

***NO:** Does not allow the use of variable names for the element.

PGM Parameter

Specifies whether the list element is a program name. PGM(*YES) is valid only for a statement-label, *CHAR, *NAME, *SNAME, *CNAME, and *GENERIC types. The specification of the PGM(*YES) parameter has no effect on the element being defined by the ELEM statement; it only indicates to the compiler that the value for this element is a program name. This information is stored so it can be included in the output of the Display Program References (DSPPGMREF) command.

***NO:** The element defined in this ELEM statement is not a program name.

***YES:** The element is a program name.

DTAARA Parameter

Specifies whether the list element is a data area name. DTAARA(*YES) is valid only for statement-label, *CHAR, *NAME, *SNAME, *CNAME, and *GENERIC types. The specification of the DTAATA(*YES) parameter has no effect on the element being defined by the ELEM statement; it only indicates to the compiler that the value for this element is a data area. This information is stored so it can be included in the output of the Display Program References (DSPPGMREF) command.

***NO:** The element defined in this ELEM statement is not a data area name.

***YES:** The element is a data area name.

FILE Parameter

Specifies whether the list element is a file name and the expected use of the file. The element can be specified as the name of a file that has a specific use so that, at compile time, the names can be used to get file reference information about where the files are used. FILE is valid only if the value for TYPE is statement-label, *CHAR, *NAME, *SNAME, *CNAME, or *GENERIC. The specification in the FILE parameter has no effect on the list element being defined by the ELEM statement; it

only indicates to the compiler that the value for this element is a file name and what type of file it is. This information is stored so it can be included in the output of the DSPPGMREF (Display Program References) command. One of the following types of files can be specified:

***NO:** The list element defined in this ELEM statement is not a file name.

***IN:** The list element is an input file name.

***OUT:** The list element is an output file name.

***UPD:** The list element is an update file name.

***INOUT:** The list element value is the name of a file to be used for both input and output.

***UNSPFD:** The list element value is the name of a file, but its use cannot be specified.

FULL Parameter

Specifies whether the number of characters in the list element must be exactly the same as the number specified in the LEN parameter (if specified) or its default length (if LEN is not specified).

***NO:** The number of characters in the list element can be less than that specified by the LEN parameter.

***YES:** The number of characters in the list element must equal the number specified by the LEN parameter or the default length for that type. The exact length is valid only for element types *LGL, *CHAR, *NAME, *SNAME, *CNAME, *GENERIC, *VARNAME, and *HEX.

EXPR Parameter

Specifies whether the list element can accept an expression containing a character concatenation. Valid character concatenation operators are as follows:

Concatenation	*CAT or, ||
Blank insertion with concatenation	*BCAT or, |>
Blank truncation with concatenation	*TCAT or, |<

***NO:** The element value cannot be a concatenation expression.

***YES:** The element value can be a concatenation expression. *YES is not valid if a value is specified for the CONSTANT parameter.

VARY Parameter

Specifies whether the list element value passed to the CPP is preceded by a length value that indicates the number of characters entered for the element's value.

***NO:** The element value is not preceded by a length value.

***YES:** The element value passed to the CPP is preceded by a 2-byte binary length field that indicates the number of characters actually specified for the list element. The data is passed in a field of the length specified by the LEN parameter or by the default length. *YES is valid only for the following element types: *CHAR, *NAME, *SNAME, *CNAME, *GENERIC, and *VARNAME. If a CL variable is specified for this element, the 2-byte binary length field contains the length of the variable value with trailing blanks removed, not the declared length of the CL variable.

PASSATR Parameter

(For IBM-supplied commands) Specifies whether an attribute byte is to be passed to the CPP with the list element data. PASSATR is not valid for TYPE(statement-label) or for TYPE(*ZEROELEM).

***NO:** No attribute byte is passed with the list element.

***YES:** An attribute byte is passed with the list element; the attribute byte indicates whether the data value came from the default, the data type of the value, and (if TYPE(*CHAR) was specified) whether the character string was enclosed in apostrophes.

DSPINPUT Parameter

Specifies whether the keyword value is shown (displayed) in the job log or on a prompt display. DSPINPUT(*PROMPT) and DSPINPUT(*NO) are valid only for the following element types: *CHAR, *NAME, *SNAME, *CNAME, *GENERIC, *DEC, *LGL, *INT2, *INT4, *DATE, *TIME, and *HEX.

Note: The DSPINPUT parameter will have no effect on the job log entries for a database reader job or for imbedded commands (for example, a command submitted on the SBMJOB command).

***YES:** The parameter value is shown both on the prompt screen and in the job log.

***PROMPT:** The parameter value is shown on the prompt screen but not in the job log.

***NO:** The parameter value is not shown in either the job log or a prompt screen. When a previously entered command is retrieved, the nondisplayed field entries must be retyped and their previous values are not retrievable. When a job log entry is created, the nondisplayed field is replaced by empty parentheses ().

CHOICE Parameter

Specifies the text that is displayed to the right of the prompt line of each parameter on the prompt screen. Up to 30 characters of text can be displayed.

***VALUES:** Each possible value is displayed in the possible values field, separated by a comma and a space. If values are specified for the Default, Single value, or Special value parameters, the first value displayed is the default value; the next value is a single value, and the values following that are special values. If there are too many values to fit in 30 characters, the last value is followed by three periods.

Examples of possible values text follow:

- If *NO is specified on the RSTD parameter and *DEC is specified on the TYPE parameter and the RANGE parameter is not specified, the word "RANGE" is displayed in the possible values field.

The resulting line will appear in the form: RANGE, *XXX, *YYY, *ZZZ...

- If *NO is specified on the RSTD parameter and *DEC is specified on the TYPE parameter and the RANGE parameter is specified, the range of possible values is displayed in the possible values field. The resulting line will appear in the form: a-b, *XXX, *YYY, *ZZZ (where a and b are numerals defining the range).
- If *YES is specified on the RSTD parameter, the possible values displayed are determined by the VALUES parameter, the SNGVAL parameter, and the SPCVAL parameter. The resulting line will appear in the form: *XXX, *YYY, *ZZZ...

***NONE:** No values are displayed.

***PGM:** A program that is called determines the values that are displayed. The program that is called is identified in CHOICEPGM parameter.

message-identifier: Specify the message ID of the message used to retrieve the message containing the text for the possible values field. The message file specified on the PMTFILE parameter of the Create Command (CRTCMD) command is used to find the message.

'choices-text': Specify no more than 30 characters, enclosed in apostrophes.

CHOICEPGM Parameter

Specifies the name of the program that is called during the prompting to fill in the possible choices text and the permissible values during prompting. This parameter must be specified only if CHOICE(*PGM) is specified.

***NONE:** No program is identified to fill in the possible choices text and permissible values.

The possible library values are:

***LIBL:** The library list is used to locate the program name.

***CURLIB:** The current library for the job is used to locate the program name. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the program name is located.

program-name: Specify the name of the program to be called during prompting to fill in the possible choices text or to supply a permissible value.

If any exception occurs when the program is called, no possible-choices text will be left blank, and the list of permissible values will be obtained from the command.

PROMPT Parameter

Specifies the prompt text, if any, that is used for the list element being defined in this ELEM statement. The prompt text describes the element to the user, who may enter a response to the information displayed. Prompt text cannot be specified if TYPE(*ZEROELEM) is specified or if a constant value is specified in the CONSTANT parameter.

***NONE:** No prompt text is displayed for the list element defined by this ELEM statement. The input field still asks for this list element, but no text is displayed with it.

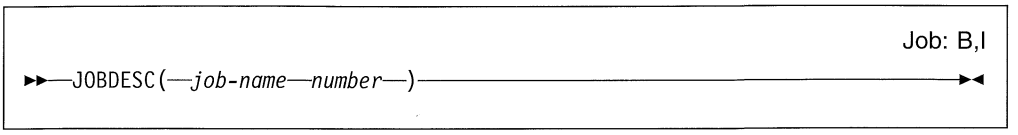
message-identifier: Specify the message identifier that identifies the prompt text message of up to 30 characters displayed when the program is asking for the list element. If a message having the specified identifier cannot be found in the message file contained in the PMTFILE parameter of the Create Command (CRTCMD) command, the message identifier itself is used as the prompt text.

'prompt-text': Specify the prompt text displayed when the program is asking for the list element. The text must be a character string of no more than 30 characters, enclosed in apostrophes.

Examples

Example 1: Defining a JOBDESC Parameter

Parameter syntax:



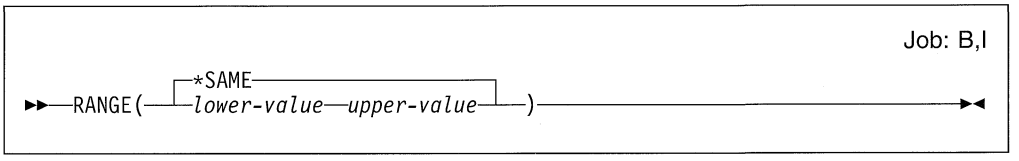
Command definition statements:

```
       PARM KWD(JOBDESC) TYPE(L1)
L1:    ELEM TYPE(*NAME) MIN(1)
       ELEM TYPE(*DEC) LEN(2) MIN(1) REL(*LE 60)
```

The parameter named JOBDESC can be omitted, but, if present, it *must* be a list of two elements. The first element is a name, and the second is a 2-digit number that is less than or equal to 60.

Example 2: Defining a RANGE Parameter

Parameter syntax:



Command definition statements:

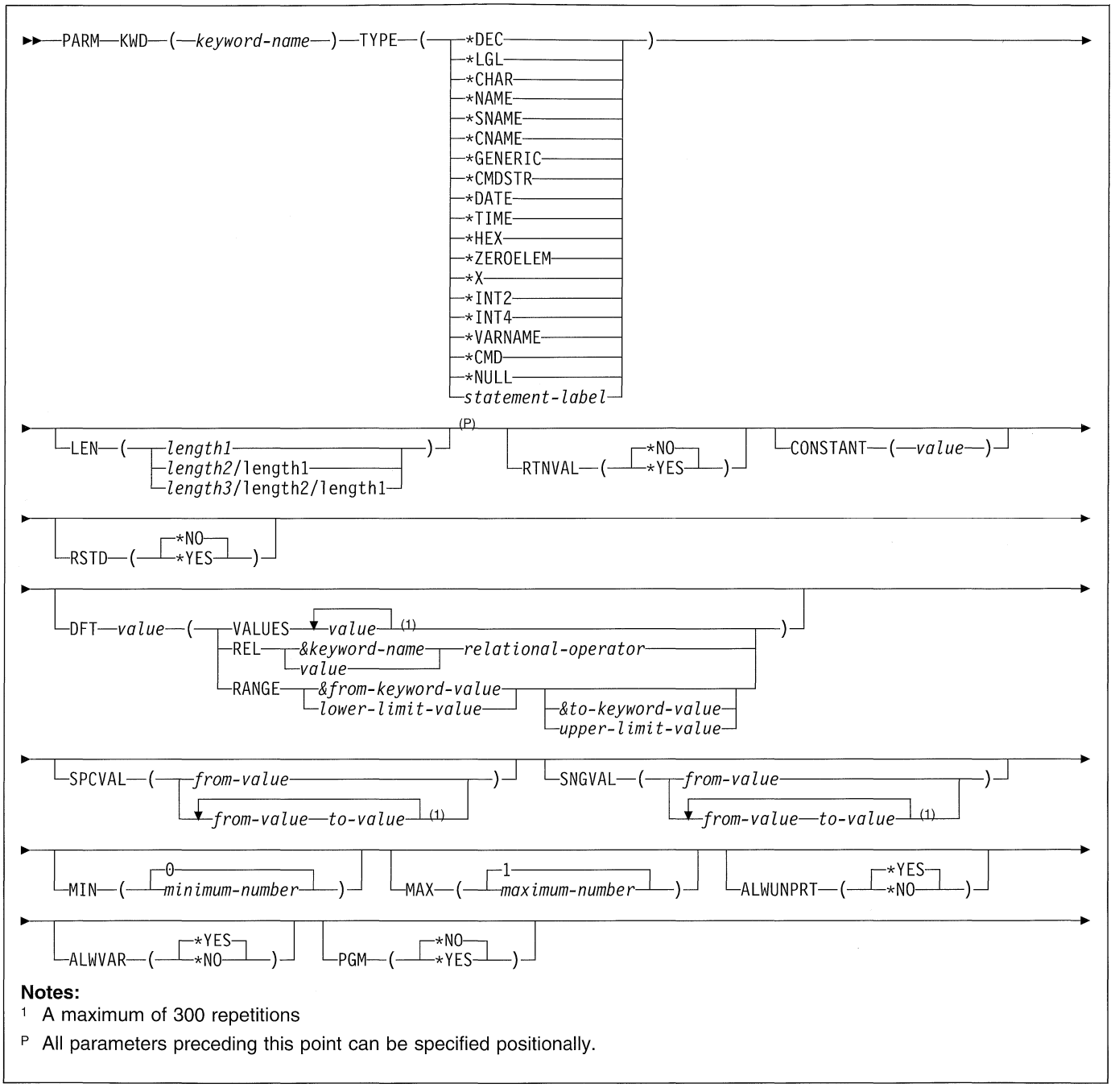
```
       PARM KWD(RANGE) TYPE(L1) DFT(*SAME) +
         SNGVAL((*SAME 101))
L1:    ELEM TYPE(*DEC) MIN(1) REL(*LE 100)
       ELEM TYPE(*DEC) MIN(1) REL(*LE 100)
```

The parameter named RANGE can be omitted, but, if present, it must be a list of two numbers, neither of which can be greater than 100. Validity checking is performed to determine that the lower-limit value is less than the upper-limit value. To allow the CPP to determine whether the value passed is a user-specified value or the default, *SAME is mapped to 101.

PARM (Parameter) Statement

The Parameter (PARM) statement defines a parameter of a command being created. A parameter is the means by which a value is passed to the command processing program (CPP). One PARM statement must be used for each parameter that appears in the command being defined. The order in which the PARM statements are entered into the source file determines the order in which the parameters must be specified when the command is entered in positional form and the order in which they are passed to the validity checker and to the CPP. A maximum of 75 parameters can be defined for one command. Note, however, that commands having a large number of parameters take longer to run, regardless of how many parameters are actually coded.

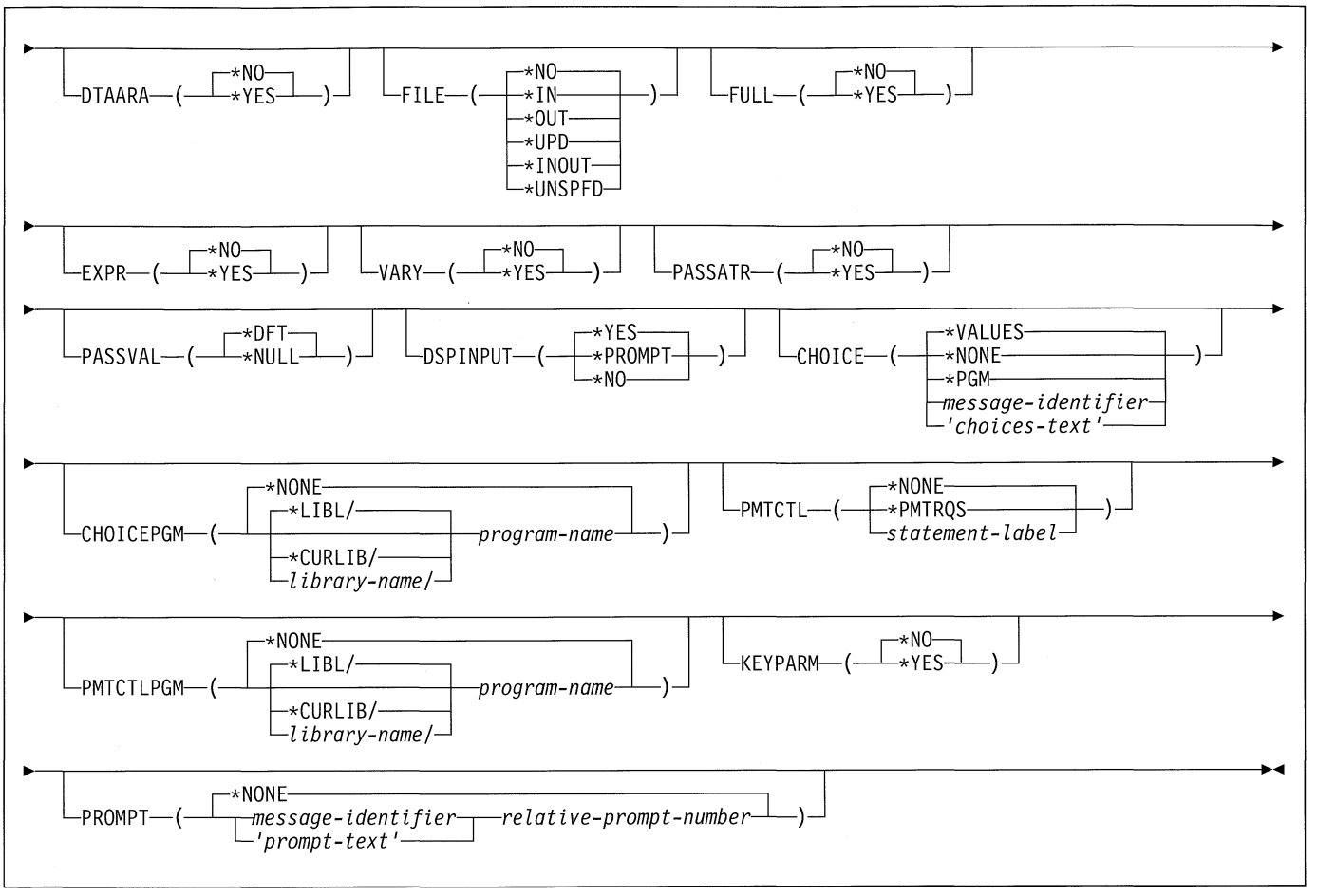
Note: The PARM statement contains certain parameters and predefined values that can be used only when IBM-supplied CPPs are called by the command being defined. Limitations in some high-level languages reduce the usefulness of these values in the definition statements of user-defined commands. These parameters and values are identified by the phrase "(*For IBM-supplied commands)*" that immediately follows the parameter keyword (if the entire parameter is for IBM-supplied commands only) or the predefined value to which it applies.



Notes:

1 A maximum of 300 repetitions

P All parameters preceding this point can be specified positionally.



KWD Parameter

Specifies the keyword name of the parameter being created. The keyword identifies the parameter in the command, and it is used when the parameter is entered in keyword form. Enter the keyword name using no more than 10 alphanumeric characters, the first character being alphabetic.

TYPE Parameter

Specifies the type of the value that can be specified for the parameter named in the KWD parameter. The value can be an integer, decimal, hexadecimal, or logical value, or a quoted or unquoted character string that can be a name, date, or time. The value can also be a command. Enter one of the following values to specify the type of parameter.

Note: For more information on names (below), see "Rules for Specifying Names" in chapter 2.

***NAME:** The list element is a character string that represents a basic name. The maximum length of the name is 256 characters; the first character must A-Z, $, @, or #. The remaining characters can be the same as the first character but can also include the numbers 0 through 9, underscores (_), and periods (.). The name can also be a string of characters starting and ending with double quotation marks ("). If a special value is used (as in *LIBL or *NONE), it must be specified in the SPCVAL parameter.

***SNAME:** The list element is a character string that represents a simple name. The maximum length of the name is 256 characters; the first character must be A-Z, $, @, or #. The remaining characters can be the same as the first but may also include the numbers 0 through 9 and underscores (_). Periods (.) are not allowed. If a special value is used (such as *LIBL or *NONE), it must be specified in the SPCVAL parameter.

***CNAME:** The list element is a character string that represents a communications name. The maximum length of the name is 256 characters; the first character must be A-Z, $, @, or #. The remaining characters can be the same as the first character but may also include the numbers 0 through 9. Periods (.) and underscores (_) are not allowed. If a special value is used (such as *LIBL or *NONE), it must be specified in the SPCVAL parameter.

***GENERIC:** The parameter value is a character string that represents a generic name. A generic name contains a maximum of 255 characters followed by an asterisk (*) or 256 characters without an asterisk and must conform to the rules for generic names. The name identifies a group of objects whose names all begin with the characters preceding the *. If an * is not included, the system assumes that the generic name is a complete object name.

***CMDSTR:** The parameter value is a command string that is checked for validity by the command analyzer. It is passed to the CPP as a character string.

***DATE:** The parameter value is a character string that represents a date. When it is passed to the CPP, it is always passed in the format *Cyymmdd*, where C = century, yy = year, mm = month, and dd = day. The century digit is set to 0 (zero) if yy is equal to or greater than 40 (years 1940 through 1999); it is set to 1 (one) if yy is less than 40 (years 2000 through 2039). When a date value is specified in this PARM statement, it must be specified in the format *mmddyy*. When a user enters a date in the command at run time, it must be entered in the format specified by the system value QDATFMT. The system value QDATSEP specifies the optional separator character that can be used when the date is entered. If the separator character is used, the date must be enclosed in apostrophes.

***TIME:** The parameter value is a character string that represents a time. It is entered in the format *hhmmss* or, if enclosed in apostrophes, in the format *'hh:mm:ss'*, where hh = hours, mm = minutes, and ss = seconds. It is passed to the CPP in a 6-byte character string as hhmmss.

***CHAR:** The parameter value is a character string that (optionally) can be enclosed in apostrophes. If the character string contains any special characters (not including an asterisk (*)), it must be enclosed in apostrophes. The maximum length of the character string is 3000 characters.

***DEC:** The parameter value is a packed decimal number.

***LGL:** The parameter value is a logical value, either a one ('1') or a zero ('0').

***HEX:** The parameter value is in hexadecimal form. The specified characters must be 0 through F. They are converted to hexadecimal (EBCDIC) characters (2 hex digits per byte), right-justified, and padded on the left with zeros. If the value is enclosed in apostrophes, an even number of digits is required. If the value is not enclosed in apostrophes, an even number of digits is *not* required.

***ZEROELEM:** The parameter is always considered as a list of zero elements, for which no value can be specified in the command. It is used to prevent a value from being entered for a parameter that is a list even though the CPP expects one. For example, if two commands use the same CPP, one command could pass a list for a parameter and the other command may not have any values to pass. The second command would be coded with TYPE(*ZEROELEM).

***X:** (For IBM-supplied commands) The parameter value is a character string, variable name, or numeric value. The value is passed as a numeric value if it contains only digits, a "+" or "–" sign, and/or a decimal point; otherwise, it is passed as a character string.

***INT2:** The parameter value is an integer that is passed as a 2-byte signed binary number. CL programs do not support binary values in variables.

***INT4:** The parameter value is an integer that is passed as a 4-byte signed binary number. CL programs do not support binary values in variables.

***VARNAME:** (For IBM-supplied commands) The parameter value is a CL variable name that is passed as a character string. The name can contain a maximum of 11 characters, including the ampersand (&).

***CMD:** (For IBM-supplied commands) The parameter value is a command. For example, the IF command has a parameter called THEN whose value must be another command. The command is checked for validity by the command analyzer.

***NULL:** The parameter value is to be a null pointer, which can be used as a constant place holder. A DEP statement or the REL and RANGE keywords of other PARM statements may not reference the value of a parameter defined with TYPE(*NULL).

statement-label: Specify a qualified name or a mixed list of values. The statement label specified here by the TYPE parameter is the statement label that identifies the first of a series of QUAL or ELEM statements that further describe the qualified name or the mixed list being defined by this PARM statement.

LEN Parameter

Specifies the length of the parameter value that is passed to the CPP. LEN is not allowed if TYPE was specified as *INT2, *INT4, *DATE, *TIME, *CMD, *ZEROELEM, *NULL, or statement-label. With other TYPE specifications, this parameter has the following applications:

- If TYPE(*DEC) was specified, the decimal length is specified in the form (length1 length2), where length1 specifies the total number of digits in the value (including the decimal portion), and length2 specifies the number of allowable decimal digits to the right of the decimal point. (The value for length2 is optional. Zero is assumed if it is not entered.)
- If TYPE(*CHAR), TYPE(*NAME), TYPE(*SNAME), TYPE(*CNAME), TYPE(*CMDSTR), or TYPE(*VARNAME) was specified, only length1 is specified. It identifies the number of characters passed.
- If TYPE(*HEX) was specified, only length1 is specified. This length specifies the number of characters passed after the hexadecimal digits have been converted to character digits. Because 2 hexadecimal digits are converted to 1 decimal digit, the number of hexadecimal digits converted is twice the value of this length.
- If TYPE(*X) was specified, the LEN parameter is used as follows:

- For character data, length1 specifies the minimum length to be passed. If a longer value is entered, the entire value is passed.
- For decimal data, length2 and length3 specify the length and decimal positions for a constant value. If a variable is entered, it is passed according to the variable attributes.
- For a logical value, length1 specifies the length of the value, which is always 1.

The default length that is assumed by the system and the maximum length for each type of parameter value are shown in the following table:

Data Type	Default Length	Maximum Length[1]
*DEC	(15 5)	(24 9)
*LGL	1	1
*CHAR	32	3000
*NAME	10	256
*SNAME	10	256
*CNAME	10	256
*GENERIC	10	256
*HEX	1	256
*X	(1 15 5)	(256 24 9)
*VARNAME	11	11
*CMDSTR	256	6000

1 The maximum length shown here is the maximum length allowed by the Command Definition. The high-level language used as the CPP for the command may have different maximum lengths for these data types (for example, *DEC values in CL programs have a maximum length of (15 9)).

Whereas the maximum shown here is the maximum for the values used in the command when it is run, the maximum length of character constants specified in the command definition is limited to 32 characters. This restriction applies to the following parameters: CONSTANT, DFT, VALUES, REL, RANGE, SPCVAL, and SNGVAL.

For data types for which length cannot be coded, the following are the maximum lengths and the lengths passed.

Data Type	Maximum Length	Length Passed
*DATE[1]	8	7
*TIME[2]	8	6
*ZEROELEM	0	0
*INT2[3]	6	2
*INT4[4]	11	4
*CMD	Any	Length needed
statement-label[5]	–	–

1 If a date is specified, the value is passed as 7 characters.
2 If a time is specified, the value is passed as 6 characters.
3 The value must meet the following condition: $-2^{15} \leq$ value $\leq 2^{15}-1$. The value is passed as a 2-byte signed binary number.
4 The value must meet the following condition: $-2^{31} \leq$ value $\leq 2^{31}-1$. The value is passed as a 4-byte signed binary number.
5 The length of the data accepted and passed is defined by the ELEM or QUAL statement that the label identifies.

RTNVAL Parameter

Specifies whether a value is returned by the CPP through the parameter being defined in this PARM statement.

***NO:** No value can be returned in the parameter being defined. The parameter is an input parameter only.

***YES:** A value is to be returned by the CPP in the parameter. A CL variable name must be specified for the parameter to receive the value (when the command is called). If a variable name is not specified when the command is invoked, a null pointer is passed to the CPP.

- *YES is valid only if the TYPE parameter was specified as *DEC, *CHAR, *LGL, or *X. Also, *YES is valid only on commands that are limited to CL programs.
- *YES is not valid with MAX(>1), CONSTANT, DFT, RSTD, VALUES, REL, RANGE, SPCVAL, SNGVAL, FILE, FULL, EXPR, or ALWVAR(*NO).
- RTNVAL(*YES) can be specified if *BPGM, *IPGM, *BREXX, or *IREXX is specified in the CRTCMD command that uses the source file containing this PARM statement.
- VARY(*YES) must be specified if PASSATR(*YES) and RTNVAL(*YES) have been specified.

CONSTANT Parameter

Specifies that a value is passed to the CPP as a constant when the command being defined is processed. The parameter does not appear externally on the command. The value specified in this parameter (if any) must satisfy the requirements specified by the TYPE, LEN, VALUES, REL, RANGE, SPCVAL, and FULL parameters. As noted in the LEN parameter chart, if a character constant is specified in this parameter, it can be no longer than 32 characters.

If a constant is specified for the parameter being defined, no prompt text can be specified for the PROMPT parameter in the PARM statement, because the parameter will not be prompted.

The CONSTANT parameter is not valid with TYPE(*CMD), TYPE(*NULL), TYPE(*ZEROELEM), MAX(>1), DFT, RTNVAL(*YES), or EXPR(*YES).

Variables cannot be coded for this parameter.

RSTD Parameter

Specifies whether the value entered for the parameter (specified in the PARM statement) is restricted to only one of the values given in the VALUES, SPCVAL, or SNGVAL parameters, or whether the value can be any value that satisfies the requirements specified by the TYPE, LEN, REL, RANGE, SPCVAL, SNGVAL, and FULL parameters.

***NO:** The value entered for the parameter specified by KWD in this PARM statement can be anything that matches the requirement specified by parameters TYPE, LEN, REL, RANGE, SPCVAL, SNGVAL and FULL in this PARM statement.

***YES:** The value entered for the parameter specified by KWD in this PARM statement is restricted to one of the values in the VALUES parameter, or to one of the from-values in the SPCVAL or SNGVAL parameters. *YES cannot be specified if TYPE(statement-label), TYPE(*CMD), TYPE(*NULL), TYPE(*ZEROELEM), TYPE(*X), or RTNVAL(*YES) is specified.

DFT Parameter

Specifies the default value that is assigned to the parameter if a value is not specified by the user. That is, the default value is used as the value of the parameter if the user omits the parameter while entering the command or if *N is entered as the parameter value. The default value must satisfy one of the following:

- It must match the requirements specified by the TYPE, LEN, REL, RANGE, and FULL parameters.
- It must be one of the from-values in the SPCVAL or SNGVAL parameters.
- If the default is a character constant, it can have no more than 32 characters (as noted in the LEN parameter chart).
- If RSTD(*YES) is specified, it must be in the list of values in the VALUES parameter or in the list of from-values of the SPCVAL or SNGVAL parameters.
- It must be a from-value in the SNGVAL parameter if the parameter being defined is a list of unlike values or it is a qualified name. This is true when a statement label is specified for TYPE; the label is used to identify a QUAL or ELEM statement.

The DFT parameter is not valid if CONSTANT is specified. The DFT parameter is valid only if MIN is 0, which means the parameter named in the KWD parameter is optional. A default cannot be specified if RTNVAL(*YES) is specified; instead, a null pointer is passed for the default. A default cannot be specified if TYPE(*CMD), TYPE(*ZEROELEM), or TYPE(*NULL) is specified. If TYPE(*VARNAME) is specified, a default special value can be specified; a default variable name cannot be specified.

If DFT is not specified and MIN(0) and RTNVAL(*NO) are specified, then the default assumed is as follows, depending on the specified type:

Assumed Default	Parameter Types
0	*DEC *INT2 *INT4
	*ZEROELEM
'0'	*LGL
zeros	*DATE *TIME *HEX
blanks	*CHAR *NAME *SNAME
	*CNAME *GENERIC
	*VARNAME *CMDSTR
null	*CMD *X *NULL

An *assumed* default value is not displayed by the command prompt; a blank input field is shown instead. If a default is specified in the DFT parameter, it is displayed by the prompt exactly as specified.

value: Specify the default value that meets the specified requirements or that is one of the values specified in the VALUES, SPCVAL, or SNGVAL parameters.

Variables cannot be coded for this value.

VALUES Parameter

Specifies a list of up to 300 constants (fixed values) from which one constant can be entered as the value of the parameter named in the KWD parameter. The VALUES parameter is valid only if all of the following are true: RSTD(*YES) is specified, both the RANGE and REL parameters are not specified, and each constant matches the attributes specified by the TYPE, LEN, and FULL parameters. As noted in the LEN parameter chart, character constants specified in this parameter can be no longer than 32 characters. Enter the constants (not more than 300) that can be specified as the value of the parameter. The VALUES parameter is not valid if TYPE(*CMD), TYPE(*CMDSTR), TYPE(*X), TYPE(*NULL), TYPE(statement-label), TYPE(*VARNAME), or TYPE(*ZEROELEM), or if RTNVAL(*YES) is specified.

REL Parameter

Specifies the relationship between the parameter value of this parameter and the value of a constant or another parameter. If a keyword is specified, it must be preceded by an ampersand (&) to indicate that it is the value of the keyword that is to be tested. The value associated with the referenced keyword, not the user-specified value, is the value passed to the CPP. If the relationship is with another parameter whose value is a list of values or a qualified name, only the first value is used in the comparison.

To specify the relationship, enter one of the following relational operators followed by either a constant or the keyword name (KWD) of the other parameter (which must be preceded by an &).

*LT	less than
*LE	less than or equal to
*EQ	equal to
*GE	greater than or equal to
*GT	greater than
*NL	not less than
*NE	not equal to
*NG	not greater than

The REL parameter is not valid if RTNVAL(*YES) is specified, if either RANGE or VALUES is specified, or if TYPE is specified as *LGL, *VARNAME, *CMD, *CMDSTR, *X, *ZEROELEM, *NULL, or a statement label.

If a character type is specified by TYPE(*CHAR), the EBCDIC value of the character string is used as an unsigned integer in the comparison. As noted in the LEN parameter chart, if a character constant is specified in this parameter, it can be no longer than 32 characters.

Variables can be coded for this element.

RANGE Parameter

Specifies the range (the limits) for the value of the parameter. The parameter value must be greater than or equal to the lower limit value specified, and it must be less than or equal to the upper limit value specified. For example, 15 would be valid if RANGE was specified as (0 16).

For nonnumeric data types, such as *CHAR, the range of values and the data specified are right-justified and padded on the left with blanks. A numeric range should not be used to define an interval for nonnumeric data unless leading zeros are specified or the data is only 1 character long.

Variables can be coded for this element.

The upper and lower limits of the range can be specified either by a keyword representing the value or by the value itself. If a keyword is specified, it must be preceded by an ampersand (&) to indicate that the value of the keyword is to be tested. The value of its parameter at the time of the check is used to determine the range. The value that is tested is the value passed to the CPP, not the user-specified value. If the keyword identifies a list of values or a qualified name, only the first value is used as the range limit. A keyword may not reference a parameter that is defined with PASSVAL(*NULL), and RANGE is not valid with PASSVAL(*NULL).

The RANGE parameter is not valid if RTNVAL(*YES) is specified, if either REL or VALUES is specified, or if TYPE is specified as *LGL, *VARNAME, *CMD, *CMDSTR, *X, *ZEROELEM, *NULL, or statement label. As noted in the LEN parameter chart, character constants specified in this parameter can be no longer than 32 characters.

Variables can be coded for this element.

SPCVAL Parameter

Specifies a list of up to 300 entries that define special values that can be entered on the parameter named in the KWD parameter. Each entry specifies a character string (a from-value) that can be entered even though it may not meet all validity checking requirements. If the entered character string matches the from-value of one of the entries, and the to-value is specified, the string is replaced with the to-value and is then passed to the CPP without further checking. If the to-value is omitted, the from-value is passed to the CPP. SPCVAL is not valid if TYPE is specified as *CMD, *CMDSTR, *X, *NULL, statement-label, or *ZEROELEM, or if RTNVAL(*YES) is specified.

The from-value is a character string, but the to-value can be anything that is passable. However, for TYPE(*DATE), the to-value must be specified unquoted in the mmddyy format. If a CL variable is used for the from-value, its type must be *CHAR. The to-value must be no longer than LEN specifies and, if TYPE is *DEC, *INT2, or *INT4, the type of the to-value must be the same; if TYPE is a character type (such as *CHAR, *LGL, or *DATE), the to-value must be a character string. As noted in the LEN parameter chart, character constants specified in this parameter can be no longer than 32 characters. If a to-value is not specified, the from-value must be passable.

If a to-value of *CURLIB is specified, the name of the current job library is passed to the CPP instead of the value *CURLIB. If the from-value is *CURLIB and no to-value is specified, or if the to-value is *CURLIB and it is enclosed in apostrophes, the value *CURLIB is passed to the CPP.

Variables cannot be coded for this element.

SNGVAL Parameter

Specifies a list of up to 300 single values that can be specified for a parameter being defined as a mixed list or as a qualified name (when a statement label is specified for TYPE), or specifies that it is to accept two or more values as defined by the MAX parameter. Any one of the single values can be used instead of a list of values or a qualified name that the parameter is defined to accept. Each entry specifies a character string (a from-value) that can be entered. If an entered character string matches the from-value of one of the entries and the to-value is specified, the data is replaced with the to-value and is passed to the CPP without further checking. If the to-value is omitted, the from-value is passed to the CPP.

The to-value (or the from-value, if the to-value is omitted) must be passable, as specified in the SPCVAL parameter. As noted in the LEN parameter chart, character constants specified in this parameter can be no longer than 32 characters. SNGVAL can be specified only if the MAX parameter is greater than 1 or TYPE is specified as a statement label of a QUAL or ELEM statement. Each single value can only substitute for a list of

values or a qualified name; it cannot be a list element or qualifier. It is passed as the first and only element of the list.

SNGVAL is not valid if RTNVAL(*YES) is specified or if TYPE is specified as *CMD, *CMDSTR, *NULL, *ZEROELEM, or *X.

If a to-value of *CURLIB is specified, the name of the current job library, instead of the value *CURLIB, is passed to the CPP. If the from-value is *CURLIB and no to-value is specified, or if the to-value is *CURLIB and it is enclosed in apostrophes, the value *CURLIB is passed to the CPP.

Variables cannot be coded for this element.

MIN Parameter

Specifies the minimum number of values that must be entered for the parameter being defined.

For a parameter that does not allow multiple like values, only zero (0) for optional and 1 for required can be specified as the minimum number of values.

Note: Required parameter statements must precede optional statements. If required parameter statements are not specified first, the system assumes that the specified parameter is optional, and the minimum number of values for required parameters is ignored.

For a parameter that allows multiple like values (because a value greater than 1 is specified for the MAX parameter), zero (0) indicates that no values need be entered; therefore, it is an *optional* parameter. A value equal to or greater than one (1) indicates the minimum number of values that must be entered for the parameter; therefore, it is a *required* parameter. The value exceeds the value specified for the MAX parameter and cannot exceed 1 for TYPE(*NULL).

0: The parameter is optional; it does not have to be entered.

minimum-number: Specify the minimum number of elements that must be specified for this parameter (named by the KWD parameter). If 1 is the assigned value, it specifies that at least one value is required for the parameter. If a number greater than 1 is specified, the parameter is a list that must have at least as many elements as the number specified.

MAX Parameter

Specifies, if this PARM statement is defining a simple list parameter, the maximum number of list elements that this list parameter can contain. If a value greater than 1 is specified, the parameter is capable of accepting multiple like values (that is, a simple list). This support is primarily intended for IBM-supplied commands. All values entered for this parameter (at the time the command is run) must satisfy the validity checking requirements specified by the other parameter values on this PARM statement.

Note: The values for a list parameter are passed consecutively, preceded by a 2-byte binary value that indicates the number of values entered in the parameter by the user. CL programs do not support the handling of binary values in variables.

1: The parameter accepts only one value; the parameter is not a list parameter.

maximum-number: Specify the maximum number of elements that the list parameter can accept. The specified maximum must be greater than or equal to the value specified in MIN, and less than or equal to 300. If the maximum is greater than 1 and TYPE is not a statement label that identifies a QUAL or ELEM statement, the parameter is a simple list of like elements (that is, each element in the list has the same requirements, such as type and length). If TYPE(statement-label) is specified and it points to the label of an ELEM or QUAL statement, MAX should only be specified greater than 1 if a list of lists or a list of qualified names is accepted. A maximum greater than 1 is not valid if TYPE is specified as *CMD, *CMDSTR, or *NULL, or if RTNVAL(*YES) or CONSTANT is specified.

ALWUNPRT Parameter

Specifies whether this PARM statement accepts the hexadecimal characters X'FF' or those in the range of X'00' to X'3F'. This parameter is valid only for TYPE(*CHAR) or TYPE(*X).

***YES:** Any characters can be passed to the CPP and sent to the display or printer.

***NO:** Unprintable characters cannot be passed to the CPP.

ALWVAR Parameter

Specifies whether to allow variable names for the parameter. If TYPE was specified as *VARNAME, *ZEROELEM, *NULL, or statement-label, ALWVAR(*NO) is not allowed.

***YES:** Variable names can be used for the parameter.

***NO:** Variable names cannot be used for the parameter.

PGM Parameter

Specifies whether this parameter element is a program name. PGM(*YES) is valid only for statement-label, *CHAR, *NAME, *SNAME, *CNAME, and *GENERIC types. The specification of the PGM(*YES) parameter does not have any effect on the parameter element being defined by the PARM statement; it only indicates to the compiler that the value for this parameter is a program name. This information is stored so it can be included in the output of the Display Program References (DSPPGMREF) command.

***NO:** The parameter defined in this PARM statement is not a program name.

***YES:** The parameter defined in this PARM statement is a program name.

DTAARA Parameter

Specifies whether the parameter is a data area name. DTAARA(*YES) is valid only for statement-label, *CHAR, *NAME, *SNAME, *CNAME, and *GENERIC types. The specification of the DTAATA(*YES) parameter does not have any effect on the parameter being defined by the PARM statement; it only indicates to the compiler that the value for this parameter is a data area. This information is stored so it can be included in the output of the Display Program References (DSPPGMREF) command.

***NO:** The parameter defined in this PARM statement is not a data area name.

***YES:** The parameter defined in this PARM statement is a data area name.

FILE Parameter

Specifies whether the parameter is a file name and the expected use of the file. The parameter can be specified as the name of a file that has a specific use so that, at compile time, the names can be used to get file reference information about where the files are used. The specification in the FILE parameter does not have any effect on the operation of the parameter being defined; it only indicates to the compiler that the value for this parameter is a file name and what type of file it is. This information is stored so it can be included in the output of the Display Program References (DSPPGMREF) command. The FILE parameter is valid only if *CHAR, *NAME, *SNAME, *CNAME, *GENERIC, or statement-label is specified for the TYPE parameter. The FILE parameter is not valid with RTNVAL(*YES). One of the following types of files can be specified:

***NO:** The parameter (named by KWD) is not a file name.

***IN:** The parameter value is an input file name.

***OUT:** The parameter value is an output file name.

***UPD:** The parameter value is an update file name.

***INOUT:** The parameter value is the name of a file that is used for both input and output.

***UNSPFD:** The parameter value is the name of a file, but its use cannot be specified.

The use of the file must match the type of file specified. For example, if *IN is specified, the file can be used only for input; if *UPD is specified, it can be used only to update existing records.

FULL Parameter

Specifies whether the number of characters in the parameter must be exactly the same as the number specified in the LEN parameter (if specified) or its default length (if LEN is not specified).

***NO:** The number of characters in the parameter can be less than that specified by the LEN parameter.

***YES:** The number of characters in the parameter must equal the number specified by LEN or the default length for that type. The exact length is valid only for the following parameter types: *LGL, *CHAR, *NAME, *SNAME, *CNAME, *GENERIC, *VARNAME, and *HEX. FULL(*YES) is valid with RTNVAL(*YES).

EXPR Parameter

Specifies whether the parameter named in the KWD parameter can accept an expression containing a character concatenation. Valid character concatenation operators are as follows:

Concatenation	*CAT or, ||
Blank insertion with concatenation	*BCAT or, |>
Blank truncation with concatenation	*TCAT or, |<

Restrictions: Expressions are not allowed on parameters where the TYPE parameter specifies *CMDSTR, *CMD, *ZEROELEM, *NULL, or statement-label.

***NO:** The parameter value cannot be a concatenation expression.

***YES:** The parameter value can be a concatenation expression.

VARY Parameter

Specifies whether the value that is passed to the CPP is preceded by a length value that indicates the number of characters entered for the command parameter.

***NO:** The parameter value is not preceded by a length value.

***YES:** The parameter value passed to the CPP is preceded by a 2-byte binary length field that indicates the number of characters actually specified for the parameter. The data is passed in a field of the length specified by LEN or by the default length. *YES is valid only for the following parameter types: *CHAR, *NAME, *SNAME, *CNAME, *GENERIC, *LGL, *VARNAME, *CMD, *CMDSTR, and *X. *YES must be specified if PASSATR(*YES) and RTNVAL(*YES) are specified. If a CL variable is specified for this parameter, the 2-byte binary length field contains the length of the variable value with trailing blanks removed, not the declared length of the CL variable.

PASSATR Parameter

(For IBM-supplied commands) Specifies whether an attribute byte is to be passed to the CPP with the parameter data.

***NO:** No attribute byte is passed with the parameter.

***YES:** An attribute byte is passed with the parameter; the attribute byte indicates whether the data value came from the default, the data type of the value, and, if TYPE(*CHAR) was specified, whether the character string was enclosed in apostrophes.

PASSVAL Parameter

Specifies whether a value is to be passed to the command processing program for this parameter. *NULL is not valid if the parameter is a constant param-

eter (a parameter in which a value has been specified for the CONSTANT parameter on the PARM statement, a parameter for which TYPE(*ZEROELEM) or TYPE(*NULL) has been specified, or is a list/qualified name defined by all constant ELEM or QUAL statements). *NULL also is not valid if RTNVAL(*YES) has been specified or if the value specified for the MIN parameter is greater than zero. A DEP statement or the REL and RANGE keywords of other PARM statements may not reference the value of a parameter defined with PASSVAL(*NULL).

***DFT:** The default value is always passed to the CPP.

***NULL:** A null pointer is passed to the CPP if the parameter is not specified.

DSPINPUT Parameter

Specifies whether the keyword value is shown in the job log or on a prompt display. DSPINPUT(*PROMPT) and DSPINPUT(*NO) are valid for the following parameter types: *CHAR, *NAME, *SNAME, *CNAME, *GENERIC, *DEC, *LGL, *INT2, *INT4, *DATE, *TIME, *HEX, and *CMDSTR.

Note: The DSPINPUT parameter has no effect on the job log entries for a database reader job or for imbedded commands (for example, a command submitted on the SBMJOB command).

***YES:** The parameter value is displayed on the prompt display and in the job log.

***PROMPT:** The parameter value is displayed on the prompt display but not in the job log.

***NO:** The parameter value is shown in neither the job log or a prompt display. When a previously entered command is retrieved, the nondisplay field entries must be retyped (their previous values are not retrievable). When a job log entry is created, the nondisplay field is replaced by empty parentheses ().

CHOICE Parameter

Specifies the text that is displayed to the right of the prompt line of each parameter on the prompt screen. Up to 30 characters of text can be displayed.

***VALUES:** Each possible value is displayed in the possible values field, separated by a comma and a space. If values are specified for the Default, Single value, or Special value parameters, the first value displayed is the default value; the next value is a single value, and the values following that are special values. If there are too many values to fit in 30 characters, the last value is followed by three periods.

Examples of possible values text follow:

- If *NO is specified on the RSTD parameter and *DEC is specified on the TYPE parameter and the RANGE parameter is not specified, the word "RANGE" is displayed in the possible values field. The resulting line will appear in the form: RANGE, *XXX, *YYY, *ZZZ...
- If *NO is specified on the RSTD parameter and *DEC is specified on the TYPE parameter and the RANGE parameter is specified, the range of possible values is displayed in the possible values field. The resulting line will appear in the form: a-b, *XXX, *YYY, *ZZZ (where a and b are numerals defining the range).
- If *YES is specified on the RSTD parameter, the possible values displayed are determined by the VALUES parameter, the SNGVAL parameter, and the SPCVAL parameter. The resulting line will appear in the form: *XXX, *YYY, *ZZZ...

***NONE:** No values are displayed.

***PGM:** A program that is called determines the values that are displayed. The program that is called is identified in CHOICEPGM parameter.

message-identifier: Specify the message ID of the message used to retrieve the message containing the text for the possible values field. The message file specified on the PMTFILE parameter of the Create Command (CRTCMD) command is used to find the message.

'choices-text': Specify no more than 30 characters, enclosed in apostrophes.

CHOICEPGM Parameter

Specifies the qualified name of the program that is called during the prompting to fill in the possible choices text and the permissible values during prompting. This parameter must be specified if CHOICE(*PGM) is specified, and it may not be specified otherwise.

***NONE:** No program is identified to fill in the possible choices text and permissible values.

The possible library values are:

***LIBL:** The library list is used to locate the program name.

***CURLIB:** The current library for the job is used to locate the program name. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the program name is located.

program-name: Specify the name of the program called during prompting to fill in the possible choices text or a permissible value.

If an exception occurs when the program is called, no possible choices text will be left blank, and the list of permissible values will be taken from the command.

PMTCTL Parameter

Specifies how prompting is controlled for this parameter. Prompting may be conditioned by another parameter, specified by a PMTCTL statement referred to by label in this parameter, or called for by pressing the F10 key.

***NONE:** Specifies that this parameter is always prompted, unless it is omitted due to selective prompting.

***PMTRQS:** Specifies that this parameter is not prompted unless:

- The user requests additional parameters.
- A value was entered for the parameter before the prompt was called.

statement-label: Specifies the label of the PMTCTL statement that is used to determine whether this parameter is prompted. The parameter is not prompted unless:

- The conditions specified on the referred to PMTCTL statement have been met.
- A value was entered for the parameter before the prompt was called.

PMTCTLPGM Parameter

Specifies the qualified name of the program called to convert the value specified for the parameter into a value used on a PMTCTL statement. This parameter is valid only on parameters that are referred to in the CTL parameter of a PMTCTL statement.

***NONE:** There is no program to convert the parameter value for prompt control statements. If the parameter is specified in a PMTCTL statement, the actual value is compared in the PMTCTL statement.

The possible library values are:

***LIBL:** The library list is used to locate the program name.

***CURLIB:** The current library for the job is used to locate the program name. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the program name is located.

program-name: Specifies the qualified name of the program called to convert the parameter value.

KEYPARM Parameter

Specifies whether the parameter is a key parameter. Key parameters can be used only if a prompt override program is specified on the PMTOVRPGM parameter of the CRTCMD or CHGCMD commands. The prompt override program overrides the command processing program (CPP) by showing only the key parameters on the initial prompt display. Values must be input for these parameters before the remaining parameters are shown. The remaining parameters are shown on the prompt display with the actual values, instead of *SAME or *PRV.

***NO:** The parameter is not a key parameter.

***YES:** The parameter is a key parameter. The following rules are followed if *YES is specified:

1. Key parameters must be placed before non-key parameters in the command definition statement.
2. Key parameters appear on the prompt display in the same order as they appear in the command definition statement.
3. Key parameters are valid only if a prompt override program is specified on the PMTOVRPGM parameter of the CRTCMD or CHGCMD commands; if a prompt override program is not specified, the key parameters are treated as non-key parameters, and a warning message is issued.
4. Key parameters must not be the only parameters in the command definition statement; if they are, a warning message is sent.

PROMPT Parameter

Specifies the prompt text, if any, that is used for the parameter named by the KWD parameter. The prompt text further describes the keyword and input field to the user, who may enter a response to the information displayed. For example, the prompt text for the TYPE parameter on the Create Library (CRTLIB) command is:

Library Type (*PROD *TEST):

When the prompt for the command is displayed, the prompt for the TYPE parameter is:

Library Type (*PROD *TEST): TYPE ***PROD**

The underscored field is the prompt input field where the user enters the value for the TYPE parameter. When the prompt is displayed, the prompt input field contains the default value (*PROD in this example). Prompt text cannot be specified if TYPE(*ZEROELEM) or TYPE(*NULL) is specified or if a constant value is specified in the CONSTANT parameter.

***NONE:** No prompt text is displayed for the parameter defined by this PARM statement. This parameter is still asked for by its keyword name, but no text is displayed with it.

message-identifier relative-prompt-number: Specify the message identifier that specifies the message that contains the prompt text of up to 30 characters that is displayed when the parameter is prompted. If a message having the specified identifier cannot be found in the message file specified in the PMTFILE parameter of the Create Command (CRTCMD) command, the message identifier itself is used as the prompt text.

Optionally, a relative prompt number can be specified with the message identifier. The relative prompt number specifies the order in which parameter keywords are to be asked for. This order affects only the order of prompting, not the order in which the parameters are passed to the CPP. Parameters having prompt numbers are asked for before parameters having no prompt numbers. Parameters having no prompt numbers and that do not have PMTCTL(*PMTRQS) coded are asked

for before parameters having no prompt numbers and that have PMTCTL(*PMTRQS) coded.

'prompt-text' relative-prompt-number: Specify the prompt text that is displayed when the parameter is prompted. The text must be a character string of no more than 30 characters, enclosed in apostrophes. An optional relative prompt number can be specified with the prompt text, the same as for the message identifier option.

Examples

```
PARM   KWD(X)  TYPE(*DEC)  LEN(2)  MIN(1)  REL(*GT 5)
```

The value for the parameter named X, a 2-digit decimal number, must be entered. The value must be greater than 5.

```
PARM   KWD(CLASS)  TYPE(*CHAR)  LEN(1)
  DFT(A)  VALUES(A B C)  RSTD(*YES)
```

The value of the parameter named CLASS must be A, B, or C, if entered. If it is not present, A is assumed.

```
PARM   KWD(MAXREC)  TYPE(*DEC)  LEN(3)
  MIN(1)  RANGE(&MINREC 999)
```

The value of the MAXREC parameter must be entered as a decimal number of 3 digits or less, with no digits to the right of the decimal point. The value must be greater than or equal to the value entered for parameter MINREC and also must be less than or equal to 999.

```
PARM   KWD(FILES)  TYPE(*NAME)  MIN(2)  MAX(5)
```

The FILES parameter is a homogeneous list that contains 2 to 5 names.

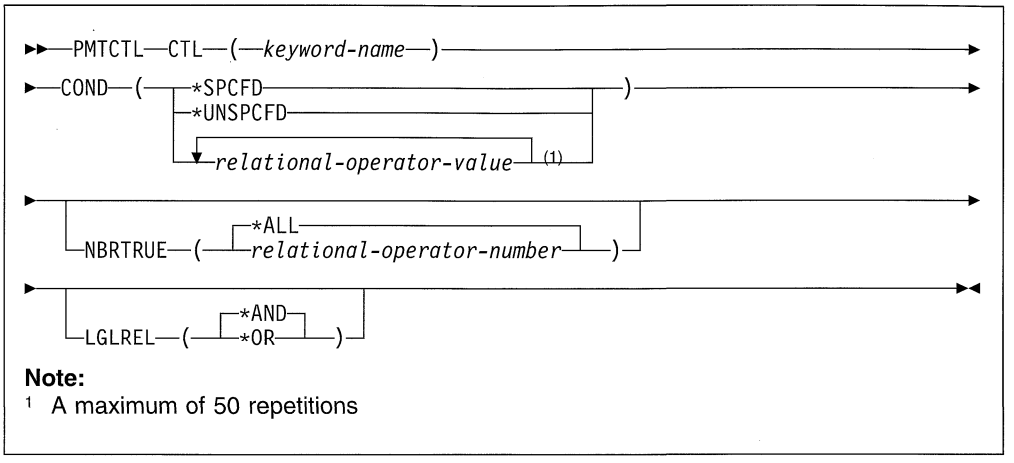
```
PARM   KWD(INVFNAME)  TYPE(*NAME)  DFT(*ALL)
  SNGVAL((*ALL XXX))  VALUES(DEPT1 DEPT2 DEPT3)
  FILE(*UPD)  MIN(0)  MAX(3)  RSTD(*YES)
  PROMPT(USR0002 1)
```

The value of the parameter named INVFNAME can be a list of up to three file names of which DEPT1, DEPT2, DEPT3, and *ALL are the valid choices. If *ALL is entered, no other values can be entered for the parameter. If this parameter is omitted, file name XXX is passed to the command processing program. If this parameter is entered through the prompt, the prompt text, as stored in the message file, is displayed and *ALL is listed as the default.

PMTCTL (Prompt Control) Statement

The Prompt Control (PMTCTL) statement specifies a condition that is tested to determine if prompting is done for the parameters whose PARM statement refers to the PMTCTL statement.

The PMTCTL statement (or the first PMTCTL statement if there is more than one) must have a statement label that matches the label referred to in the PMTCTL parameter of one or more PARM statements in the command definition source.



Note:
1 A maximum of 50 repetitions

CTL Parameter

Specifies the name of the parameter that controls the prompting. The value of the parameter specified here is compared to the value specified in the COND parameter. If the PMTCTLPGM parameter is coded for the parameter specified here, the value returned by the program specified in the PMTCTLPGM parameter is compared to the values specified in the COND parameter. If the parameter specified here is a list or qualified name, only the first list element or qualifier is compared.

COND Parameter

Specifies the condition against which the parameter specified in the CTL parameter is tested. Up to 50 conditions can be specified.

***SPCFD:** The condition is true, including the default value, if it is specified for the control parameter.

***UNSPCFD:** The condition is true only if the control parameter is not specified. It is not true if the default value is specified.

relational-operator-value: Specify the relational operator and value used to compare the value of the control parameter to the value specified in the COND parameter. Valid values are *GT, *EQ, *NL, *LT, *NE, *LE, and *NG.

NBRTRUE Parameter

Specifies the number of conditions (indicated in the COND parameter) that must be true if the parameter is prompted.

***ALL:** All the conditions must be true.

relational-operator-value: Specify the relational operator and number used to compare the number of conditions that are true to the number specified in the NBRTRUE parameter. Valid values are *GT, *EQ, *GE, *NL, *LT, *NE, *LE, and *NG.

LGLREL Parameter

Specifies, when multiple PMTCTL statements are in a group, the logical relationship of the statement to the previous statements in the group. This parameter is not allowed on the first PMTCTL statement in a group.

***AND:** This statement is in an AND relationship of the statement to the previous statements in the group.

***OR:** This statement is in an OR relationship with the previous PMTCTL statement or statements. Statements in an OR relationship are evaluated after AND relationships are evaluated.

Example

```
A: PMTCTL  CTL(TYPE)  COND((*EQ *) (*EQ *LIST))  NBTRUE(*EQ 1)
```

If TYPE(*) or TYPE(*LIST) is specified, the parameters which reference this PMTCTL statement are selected for prompting.

```
B: PMTCTL  CTL(P1)  COND((*EQ *ALL)
   PMTCTL  CTL(P1)  COND((*EQ *SOME) LGLREL(*OR)
   PMTCTL  CTL(P2)  COND((*EQ *ALL) LGLREL(*AND)
   PMTCTL  CTL(P1)  COND((*EQ *NONE) LGLREL(*OR)
   PMTCTL  CTL(P2)  COND((*NE *ALL) LGLREL(*AND)
```

The parameters which refers to this group of PMTCTL statements are selected for prompting if any of the following conditions exist:

- *ALL is specified for P1.
- *SOME is specified for P1 and *ALL is specified for P2.
- *NONE is specified for P1 and *ALL is not specified for P2.

QUAL (Qualifier) Statement

The Qualifier (QUAL) statement describes one part of a qualified name. If a name is the allowed value of a parameter or list element defined in a PARM or ELEM statement, it can be changed to a qualified name by using a QUAL statement for each qualifier used to qualify the name.

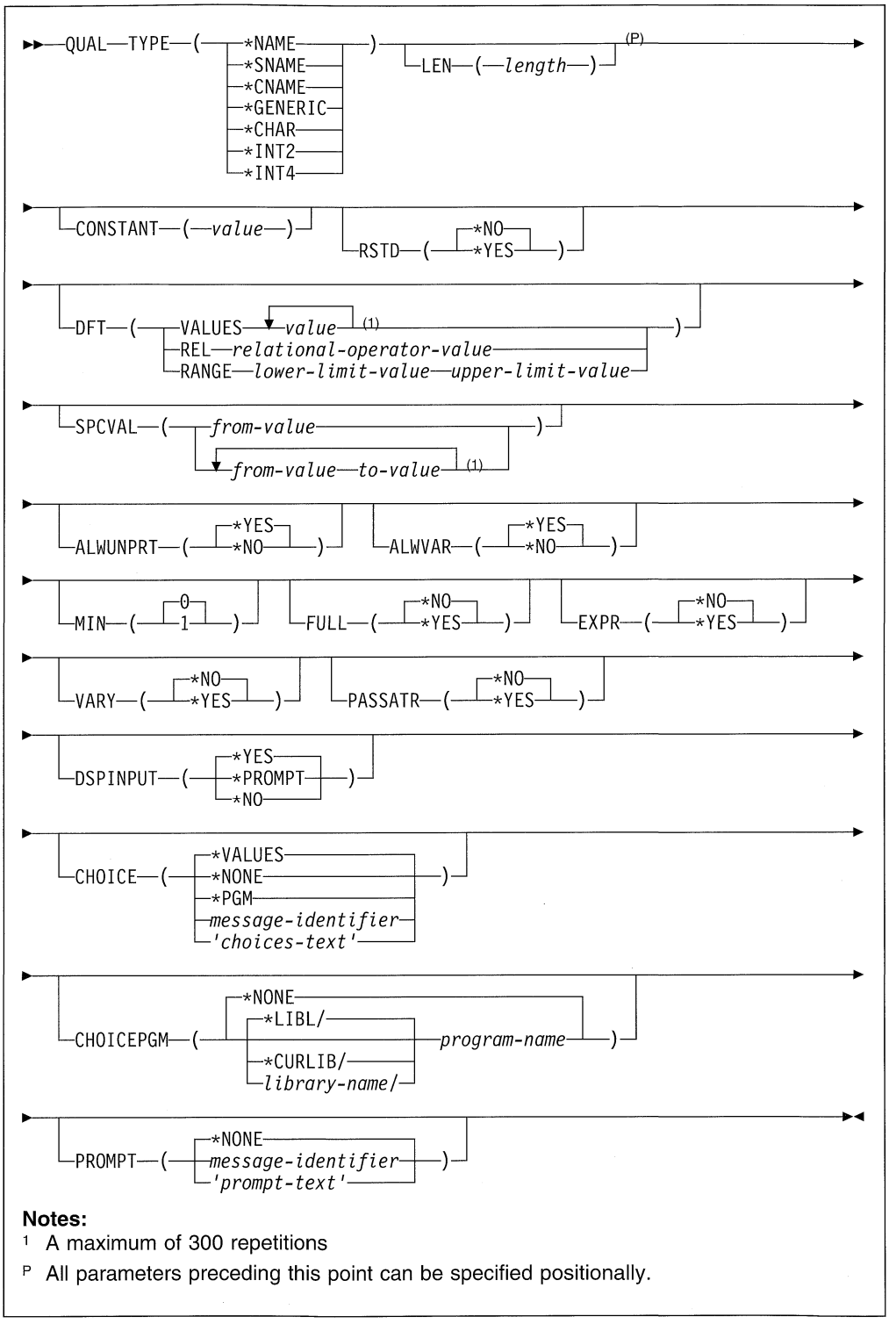
The order in which the QUAL statements are entered into the source file determines the positional order in which the qualifiers must be specified and passed to the validity checker and the command processing program (CPP).

The QUAL statement (or only the first QUAL statement if there is more than one) *must* have a statement label that matches the statement label value that must be specified in a PARM or ELEM statement for which the qualifier is being defined. The qualifiers for the parameter or list element are then entered on the command in the form value1/value2/value3 where values 1 through 3 are qualifiers described by a QUAL statement.

Note: In the System/38 environment, the values are passed in sequential order with periods as delimiters. An example of this form is value1.value2.value3.

The values are passed with each value padded to its defined length and with the delimiters (slashes or periods) removed.

Note: The QUAL statement contains certain parameters and predefined values that can be used only when IBM-supplied CPPs are called by the command being defined. Limitations in some high-level languages reduce the usefulness of these values in the definition statements of user-defined commands. These parameters and values are identified by the phrase "(*For IBM-supplied commands)*" that immediately follows the parameter keyword (if the entire parameter is for IBM-supplied commands only) or the predefined value to which it applies.



Notes:

1 A maximum of 300 repetitions

P All parameters preceding this point can be specified positionally.

TYPE Parameter

Specifies the type of qualifier used to qualify a parameter name or list element name. The qualifier can be a name or generic name, a quoted or unquoted character string, or an integer. Enter one of the following options to specify the type of qualifier. The first qualifier for any qualified name must have a type of name (*NAME) or generic name (*GENERIC).

***NAME:** The qualifier is a character string that represents a name. The maximum length of the name is 256 characters. The first character must be alphabetic (including the special characters $, #, or @), and the remaining characters must be alphanumeric, a period, or an underscore. The name must be a string of characters starting and ending with double quotation marks ("). If a special value is used (as in *LIBL or *NONE), it must be specified in the SPCVAL parameter.

***SNAME:** The qualifier is a character string that represents a name. The maximum length of the name is 256 characters. The first character must be alphabetic (including the special characters $, @, or #), and the remaining characters must be A through Z, 0 through 9, or $, #, or @. Periods (.) are not allowed. If a special value is used (as in *LIBL or *NONE), it should be specified in the SPCVAL parameter.

***CNAME:** The qualifier is a character string that represents a name. The maximum length of the name is 256 characters. The first character must be alphabetic (including the special characters $, #, or @), and the remaining characters must be A through Z, 0 through 9, $, #, or @. Periods (.) and underscores (_) are not allowed. If a special value is used (as in *LIBL or *NONE), it must be specified in the SPCVAL parameter.

***GENERIC:** The qualifier is a character string that represents a generic name. A generic name contains a maximum of 255 characters followed by an asterisk (*) or 256 characters without an asterisk. The name identifies a group of objects whose names all begin with the characters preceding the *. If an asterisk (*) is not included, the system assumes that the generic name is a complete object name.

***CHAR:** The qualifier is a character string that can (optionally) be enclosed in apostrophes. If the character string contains any special characters (not including an asterisk (*)), it must be enclosed in apostrophes. The maximum length of the character string is 3000 characters.

***INT2:** The qualifier is an integer that is passed as a 2-byte signed binary number. CL programs do not support binary values in variables.

***INT4:** The qualifier is an integer that is passed as a 4-byte signed binary number. CL programs do not support binary values in variables.

LEN Parameter

Specifies the length of the qualifier, if its data type is *NAME, *GENERIC, or *CHAR. The default length that is assumed by the system and the maximum length for each type of qualifier are shown in the following table:

Data Type	Default Length	Maximum Length[1]
*NAME	10	256[2]
*SNAME	10	256[2]
*CNAME	10	256[2]
*GENERIC	10	256[2]
*CHAR	32	3000[2]
*INT2[3]	6	
*INT4[4]	11	

1 The maximum length shown here is the maximum length allowed by Command Definition. The high-level language used as the CPP for the command may have a different maximum lengths for these data types (for example, *DEC values in CL programs have a maximum length of (15 9)).

2 Whereas the maximum shown here is the maximum for values used in the command when it is run, the maximum length of character constants specified in the command definition is limited to 32 characters. This restriction applies to the following parameters: CONSTANT, DFT, VALUES, REL, RANGE, and SPCVAL.

3 For *INT2, length cannot be specified; its assumed length is 6. The value must meet the following condition: $-2^{15} \leq$ value $\leq 2^{15}-1$. The value is passed as a 2-byte signed binary number.

4 For *INT4, length cannot be specified; its assumed length is 11. The value must meet the following condition: $-2^{31} \leq$ value $\leq 2^{31}-1$. The value is passed as a 4-byte signed binary number.

CONSTANT Parameter

Specifies that a value is passed to the CPP as a constant for the qualifier when the command being defined is processed; the qualifier does not appear externally on the command. If specified, the value must satisfy the requirements specified by the TYPE, LEN, VALUES, REL, RANGE, and SPCVAL parameters. (As noted in the LEN parameter chart, if a character constant is specified in this parameter, it can be no longer than 32 characters.)

If a constant is specified in this QUAL statement, and other QUAL statements immediately follow it, they must also be defined as constants, unless a label precedes one of them. A label indicates the beginning of a new group of QUAL statements, which can be defined differently.

Also, if a constant is specified for the qualifier being defined, no prompt text can be specified for the PROMPT parameter of this QUAL statement. However, any other qualifiers or groups of qualifiers are still prompted, and their values are still passed to the CPP as a qualified name.

The CONSTANT parameter is not valid if the DFT parameter is specified or if *YES is specified on the EXPR parameter.

Variables cannot be coded for this parameter.

RSTD Parameter

Specifies whether the value entered for the qualifier is restricted to only one of the values given in the VALUES or SPCVAL parameters, or whether any value can be used that satisfies the requirements specified by the TYPE, LEN, REL, RANGE, and SPCVAL parameters.

***NO:** The value entered for the qualifier defined by this QUAL statement can be anything that satisfies the requirements specified by the TYPE, LEN, REL, RANGE, and SPCVAL parameters in this QUAL statement.

***YES:** The value entered for the qualifier defined by this QUAL statement is restricted to one of the values in the VALUES parameter, or to one of the from-values in the SPCVAL parameters.

DFT Parameter

Specifies the default value assigned to the qualifier if a value is not specified by the user. The default value must satisfy one of the following:

- It must match the qualifier requirements specified by the TYPE, LEN, REL, and RANGE parameters.
- It must be one of the from-values in the SPCVAL parameter.
- If RSTD(*YES) is specified, it must be in the list of values in the VALUES parameter or in the list of from-values in the SPCVAL parameter.
- If the default is a character constant, it can have no more than 32 characters as noted in the LEN parameter chart.

The DFT parameter is valid only if the MIN parameter is 0, which means the qualifier defined by this QUAL statement for this list is optional. A default is not meaningful on this QUAL statement if it is the first one (defining the first part) for a qualified name and if a default is specified on the PARM or ELEM statement that this QUAL statement further defines.

If DFT is not specified, it has a default of its own: a blank (b) if TYPE was specified as *CHAR, *NAME, *SNAME, *CNAME, or *GENERIC; or a zero (0) if TYPE was specified as *INT2 or *INT4. An assumed default value is not displayed by the command prompt; a blank input field is shown instead. If a default is specified in the DFT parameter, it is displayed by the prompt exactly as specified.

The DFT parameter is not valid if the CONSTANT parameter is specified.

value: Specify the default value that meets the specified requirements or that is one of the values specified in the SPCVAL or VALUES parameters.

Variables cannot be coded for this value.

VALUES Parameter

Specifies a list of up to 300 constants (fixed values) from which one constant can be entered as the value of the qualifier. The VALUES parameter is valid only if all of the following are true: RSTD(*YES) is specified, both RANGE and REL are *not* specified, and the constant matches the attributes specified by the TYPE and LEN parameters in this QUAL statement. As noted in the LEN parameter chart, character constants specified in this parameter can be no longer than 32 characters. Enter the constants (not more than 300) that can be specified as the value of the qualifier.

REL Parameter

Specifies the relationship between the qualifier value and the value of another parameter or constant. To specify the relationship, enter one of the following relational operators followed by a constant or the value of another parameter.

*LT	less than
*LE	less than or equal to
*EQ	equal to
*GE	greater than or equal to
*GT	greater than
*NL	not less than
*NE	not equal to
*NG	not greater than

The REL parameter is not valid if either RANGE or VALUES is specified. If a character type is specified by TYPE(*CHAR), the EBCDIC value of the character string is used as an unsigned integer in the comparison. As noted in the LEN parameter chart, if a character constant is specified in this parameter, it can be no longer than 32 characters.

RANGE Parameter

Specifies the range (limits) for the value of the qualifier. The qualifier value must be greater than or equal to the lower limit value specified, and it must be less than or equal to the upper limit value specified. For example, 15 would be valid if the RANGE parameter was specified as (0 16). For nonnumeric data types, such as *CHAR, the range of values and the data specified is right-justified and padded on the left with blanks. A numerical range should not be used to define an interval for nonnumeric data unless leading zeros are specified or the data is only 1 character in length. The RANGE parameter is not valid if either the REL or VALUES parameter is specified. As noted in the LEN parameter chart, character constants specified in this parameter can be no longer than 32 characters.

SPCVAL Parameter

Specifies a list of up to 300 entries that define special values that can be entered on the parameter named in the KWD parameter on the PARM statement. Each entry specifies a character string (a from-value) that can be entered even though it may not meet all validity checking requirements. If the entered character string matches the from-value of one of the entries, and the to-value is specified, the string is replaced with the to-value and is then passed to the CPP without further checking. If the to-value is omitted, the from-value is passed to the CPP. The from-value is a character

string, but the to-value can be anything that is passable. If a CL variable is used for the from-value, its type must be *CHAR.

However, the first qualifier can only have special to-values with the from-values that are a name, a generic name, or an asterisk, followed by a name, such as *ALL.

Each to-value must be passable to the CPP. The to-value must be no longer than the LEN parameter specifies and, if TYPE is *INT2 or *INT4, the type of the to-value must be the same; if TYPE is a character type (such as *CHAR or *NAME), the to-value must be a character string. As noted in the LEN parameter chart, character constants specified in this parameter can be no longer than 32 characters. If a to-value is not specified, the from-value must be passable.

If a to-value of *CURLIB is specified, the name of the current job library is passed to the CPP instead of the value *CURLIB. If the from-value is *CURLIB and no to-value is specified, or if the to-value is *CURLIB and it is enclosed in apostrophes, the value *CURLIB is passed to the CPP.

Variables cannot be coded for this value.

ALWUNPRT Parameter

Specifies whether this QUAL statement should accept the hexadecimal characters X'FF' or those in the range of X'00' to X'3F'. This parameter is valid only for TYPE(*CHAR) or TYPE(*X).

***YES:** Any characters can be sent to the display or printer.

***NO:** Unprintable characters cannot be passed to the command processing program.

ALWVAR Parameter

Specifies whether to allow variable names for the qualifier. If TYPE was specified as *VARNAME, *ZEROELEM, *NULL, or statement-label, ALWVAR(*NO) is not allowed.

***YES:** Variable names can be used for the qualifier.

***NO:** Variable names cannot be used for the qualifier.

MIN Parameter

Specifies whether the qualifier being defined in this QUAL statement is required or optional. If MIN is not specified, 0 is assumed, which means the qualifier is optional. If a required qualified name is needed, MIN(1) must be coded on both the first QUAL statement and on the PARM or ELEM statement that references it.

0: The qualifier is optional on the name being qualified.

1: The qualifier is required on the name being qualified; it must be entered.

FULL Parameter

Specifies whether the number of characters in the qualifier value must be exactly the same as the number specified in the LEN parameter (if specified) or its default length (if LEN is not specified).

***NO:** The number of characters in the qualifier value can be less than that specified by the LEN parameter.

***YES:** The number of characters in the qualifier value must equal the number specified by the LEN parameter or the default length for that type. The exact length is valid only for the *CHAR, *NAME, and *GENERIC qualifier types.

EXPR Parameter

Specifies whether the qualifier can accept an expression containing a character concatenation. Valid character concatenation operators are as follows:

Concatenation	*CAT or, ||
Blank insertion with concatenation	*BCAT or, |>
Blank truncation with concatenation	*TCAT or, |<

***NO:** The qualifier value cannot be a concatenation expression.

***YES:** The qualifier value can be a concatenation expression.

VARY Parameter

Specifies whether the qualifier value passed to the CPP is preceded by a length value that indicates the number of characters entered for the qualifier's value.

***NO:** The qualifier value is not preceded by a length value.

***YES:** The qualifier value passed to the CPP is preceded by a 2-byte binary length field that indicates the number of characters actually specified for the qualifier. The data is passed in a field of the length specified by LEN or by the default length. *YES is valid only for the following qualifier types: *CHAR, *NAME, *SNAME, *CNAME, or *GENERIC. If a CL variable is specified for this qualifier, the 2-byte binary length field contains the length of the variable value with trailing blanks removed, not the declared length of the CL variable.

PASSATR Parameter

(For IBM-supplied commands) Specifies whether an attribute byte is to be passed to the CPP with the qualifier.

***NO:** No attribute byte is passed with the qualifier.

***YES:** An attribute byte is passed with the qualifier; the attribute byte indicates whether the data value came from the default, the data type of the value, and, if TYPE(*CHAR) was specified, whether the character string was enclosed in apostrophes.

DSPINPUT Parameter

Specifies whether the keyword value is shown in the job log or on a prompt screen.

Note: The DSPINPUT parameter will have no effect on the job log entries for a database reader job or for imbedded commands (for example, a command submitted on the SBMJOB command).

***YES:** The parameter value is shown on the prompt display and in the job log.

***PROMPT:** The parameter value is shown on the prompt display but not in the job log.

***NO:** The parameter value is not shown either in the job log or on a prompt display. When a previously entered command is retrieved, the nondisplay field entries must be retyped (their previous values are not retrievable). When a job log entry is created, the nondisplay field is replaced by empty parentheses ().

CHOICE Parameter

Specifies the text that is displayed to the right of the prompt line of each parameter on the prompt screen. Up to 30 characters of text can be displayed.

***VALUES:** Each possible value is displayed in the possible values field, separated by a comma and a space. If values are specified for the Default, Single value, or Special value parameters, the first value displayed is the default value; the next value is a single value, and the values following that are special values. If there are too many values to fit in 30 characters, the last value is followed by three periods.

Examples of possible values text follow:

- If *NO is specified on the RSTD parameter and *DEC is specified on the TYPE parameter and the RANGE parameter is not specified, the word "RANGE" is displayed in the possible values field. The resulting line will appear in the form: RANGE, *XXX, *YYY, *ZZZ...
- If *NO is specified on the RSTD parameter and *DEC is specified on the TYPE parameter and the RANGE parameter is specified, the range of possible values is displayed in the possible values field. The resulting line will appear in the form: a-b, *XXX, *YYY, *ZZZ (where a and b are numerals defining the range).
- If *YES is specified on the RSTD parameter, the possible values displayed are determined by the VALUES parameter, the SNGVAL parameter, and the SPCVAL parameter. The resulting line will appear in the form: *XXX, *YYY, *ZZZ...

***NONE:** No values are displayed.

***PGM:** A program that is called determines the values that are displayed. The program that is called is identified in CHOICEPGM parameter.

message-identifier: Specify the message ID of the message used to retrieve the message containing the text for the possible values field. The message file specified on the PMTFILE parameter of the Create Command (CRTCMD) command is used to find the message.

'choices-text': Specify no more than 30 characters, enclosed in apostrophes.

CHOICEPGM Parameter

Specifies the qualified name of the program that is called during the prompting to fill in the possible choices text and the permissible values during prompting. This parameter must be specified if CHOICE(*PGM) is specified, and may not be specified otherwise.

***NONE:** No program is identified to fill in the possible choices text and permissible values.

The possible library values are:

***LIBL:** The library list is used to locate the program name.

***CURLIB:** The current library for the job is used to locate the program name. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the program name is located.

program-name: Specify the name of the program to be called during prompting to fill in the possible choices text or a permissible value.

If an exception occurs when the program is called, no possible choices text is left blank, and the list of permissible values is taken from the command.

PROMPT Parameter

Specifies the prompt text, if any, that is used for the qualifier (defined in this QUAL statement). The PROMPT parameter is not allowed for the first qualifier or for a qualifier for which the CONSTANT parameter is specified. The prompt text for the first qualifier comes from the PARM or ELEM statement PROMPT parameter that points to the qualifier. The prompt text describes the qualifier input field to the user, who may enter a response to the information displayed.

***NONE:** No prompt text is shown for the qualifier defined by this QUAL statement. This qualifier is still prompted by an input field, but no text is shown with it.

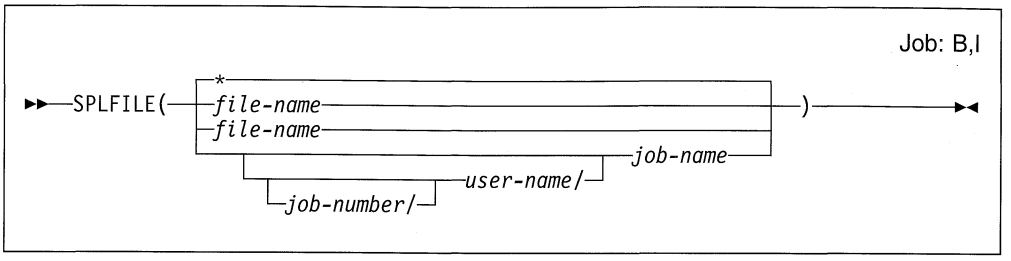
message-identifier: Specify the message identifier that specifies the message containing the prompt text of up to 30 characters that is shown. When the program is prompting for the qualifier. If a message having the specified identifier cannot be found in the message file specified in the PMTFILE parameter of the Create Command (CRTCMD) command, the message identifier itself is used as the prompt text.

'prompt-text': Specify the prompt text that is shown when the program is prompting the qualifier. The text must be a character string of no more than 30 characters, enclosed in apostrophes.

Examples

Example 1: Defining a SPLFILE Parameter

Parameter syntax:



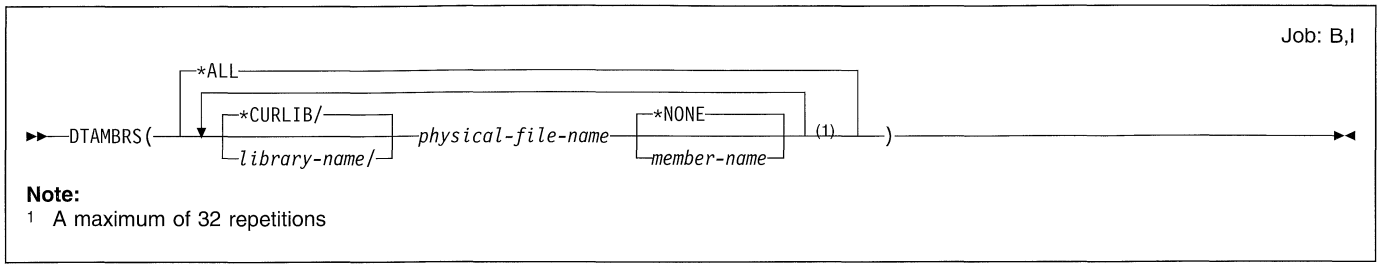
Command definition statements:

```
       PARM KWD(SPLFILE) TYPE(L1) DFT(*) SNGVAL(*)
L1:    ELEM TYPE(*NAME) MIN(1)    /*For file name */
       ELEM TYPE(Q1)  /*For job name */
Q1:    QUAL TYPE(*NAME) MIN(1)    /*For job name */
       QUAL TYPE(*NAME)     /*For user name */
       QUAL TYPE(*CHAR) LEN(6)    /*For job number */
```

The SPLFILE parameter is optional and, if not specified, defaults to an asterisk (*). Otherwise, the value consists of a two-element list. The first element is a file name and it is required. The second element is a qualified job name. The first qualifier is required; the last two qualifiers are optional.

Example 2: Defining a DTAMBRS Parameter

Parameter syntax:



Note:
1 A maximum of 32 repetitions

Command definition statements:

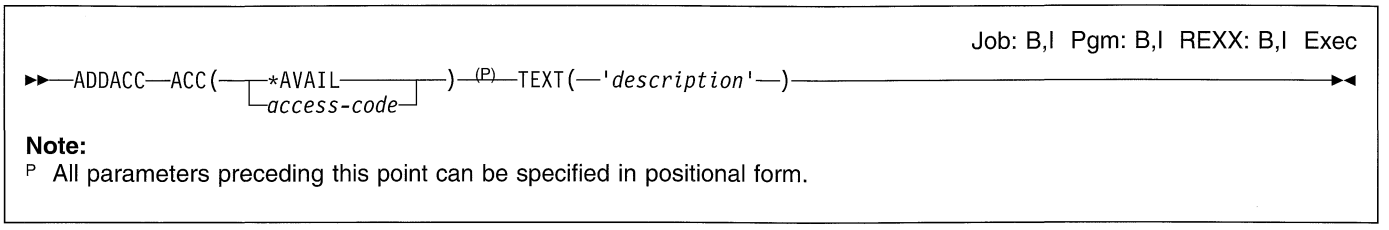
```
       PARM KWD(DTAMBRS) TYPE(L1) DFT(*ALL) MAX(32) +
         SNGVAL(*ALL)
L1:    ELEM TYPE(Q1) MIN(1)
       ELEM TYPE(*NAME) MIN(0) MAX(32) SNGVAL(*NONE) +
         DFT(*NONE)
Q1:    QUAL TYPE(*NAME) MIN(1)
       QUAL TYPE(*NAME) DFT(*CURRENT) SPCVAL(*CURRENT)
```

The parameter named DTAMBRS is optional and, if not specified, defaults to *ALL. Otherwise, the value consists of a list, each element of which is itself a list. Each sublist consists of a qualified file name optionally followed by one or more member names. If no member name is specified, *NONE is taken as the default. If no library qualifier is specified for the physical file, *CURRENT is taken as the default. This means that the library is the one currently indicated by the qualified physical file name saved in the description of the logical file to which this DTAMBRS parameter applies. Each sublist can contain one file name and up to 32 member names. Up to 32 such sublists can appear as the value of DTAMBRS.

Part 3. OS/400 CL Commands

Chapter 6. OS/400 Command Descriptions

ADDACC (Add Access Code) Command



Note:
P All parameters preceding this point can be specified in positional form.

Purpose

The Add Access Code (ADDACC) command defines a new access code to the system. Subsequent use of the new code is allowed when a document is filed either during a document interchange session or upon the use of the Add Document Library Authority (ADDDLOAUT), Edit Document Library Object Authority (EDTDLOAUT), Remove Document Library Object Authority (RMVDLOAUT), Grant Access Code Authority (GRTACCAUT), Revoke Access Code Authority (RVKACCAUT), or Remove Access Code (RMVACC) commands. The ADDACC command identifies both the added access code and the descriptive text associated with it.

Restriction: This command is shipped with public *EXCLUDE authority. Use of this command is restricted to the security officer, users with *ALLOBJ (all object) authority, and users to whom authority has been granted by using the Grant Object Authority (GRTOBJAUT) command.

Required Parameters

ACC

Specifies the access code being added to the system. The access code is a decimal number ranging from 1 through 2047, and it must not be currently defined on the system.

***AVAIL:** The system chooses the next free access code and adds it to the system; the user must specify the text. The access code selected is returned in a completion message.

access-code: Specify a decimal number ranging from 1 through 2047.

TEXT

Specifies the user-defined text that briefly describes the access code. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Examples

Example 1: Adding an Access Code

```
ADDACC  ACC(700)  TEXT('programmers')
```

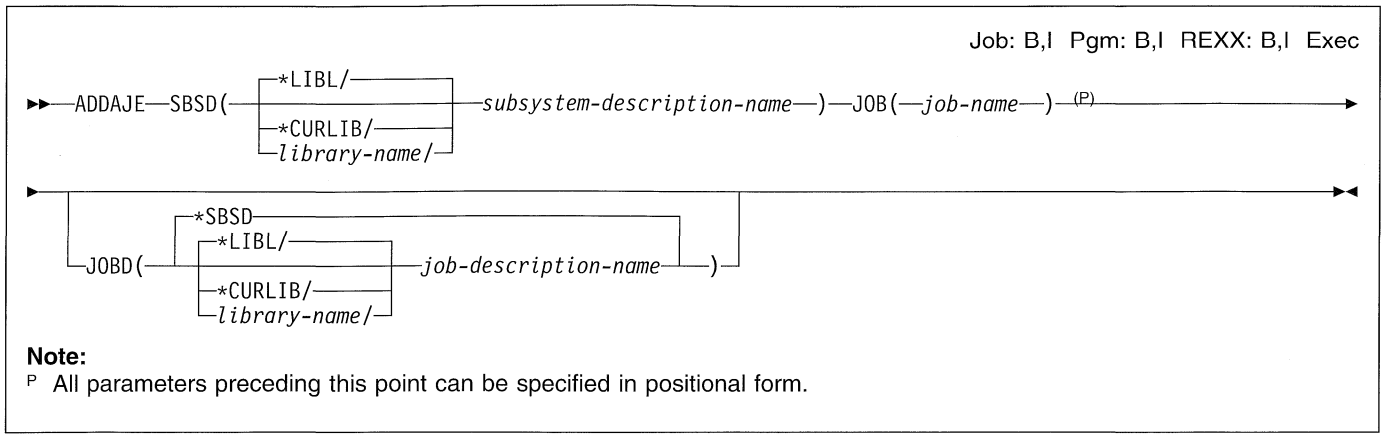
This command adds access code 700 to the system. This access code is authorized to all programmers (after the GRTACCAUT command is run). It is used when filing documents to which all programmers may have access.

Example 2: Adding Next Available Access Code

```
ADDACC  ACC(*AVAIL)
  TEXT('department managers')
```

This command adds the next available access code to the system. This access code is authorized to all department managers (after the GRTACCAUT command is run). It is used when placing in the document library objects to which all department managers may have access. The system returns a message containing the access code that was being used.

ADDAJE (Add Autostart Job Entry) Command



Note:

| (P) All parameters preceding this point can be specified in positional form.

Purpose

The Add Autostart Job Entry (ADDAJE) command adds a job entry which becomes active when the subsystem is started (the subsystem specified on the subsystem description). The user must first specify this command, then start the subsystem. The job entry identifies the job and its associated job description to the subsystem. These jobs are automatically started when the subsystem is started.

Restriction: To use this command, the user must have object operational and object management authorities for the specified autostart job entry.

Required Parameters

SBSD

Specifies the qualified name of the subsystem description where the job entry that automatically starts is added.

| The name of the subsystem description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

subsystem-description-name: Specify the name of the subsystem description where the job entry is added.

JOB

Specifies the simple name of the job that is automatically started when a subsystem is started by using the subsystem description specified in the SBSD parameter.

Optional Parameters

JOBD

Specifies the name of the job description used. If the job description does not exist when the entry is changed, a library qualifier must be specified because the qualified job description name is retained in the subsystem description. More information on this parameter is in | Appendix A, "Expanded Parameter Descriptions."

***SBSD:** The job description that has the same qualified name as the subsystem description, specified by the SBSD parameter, is used for the job being started.

| The name of the job description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

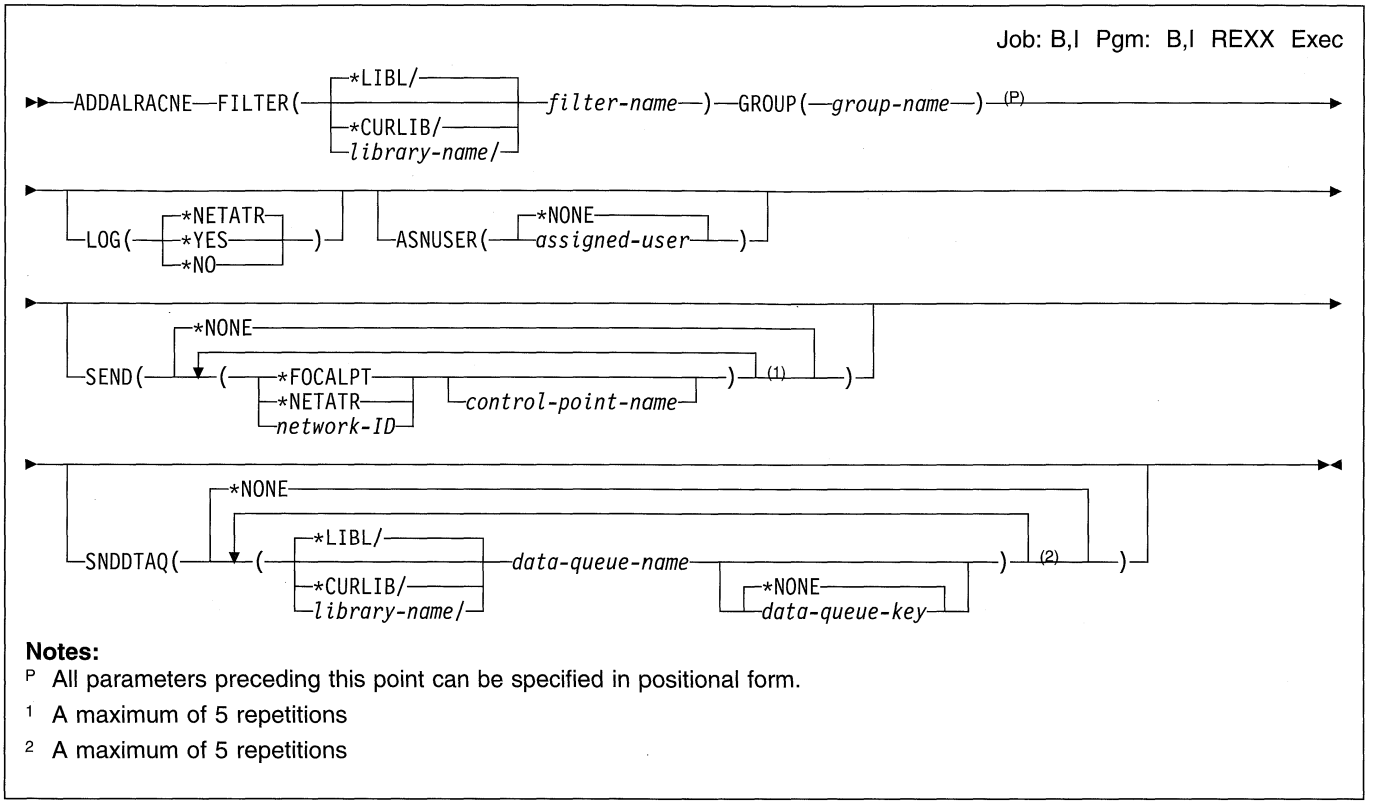
job-description-name: Specify the name of the job description that is used for the job started by this job entry.

Example

```
ADDAJE   SBSD(ACCTLIB/ACCTINT)
  JOB(ACCTINIT)  JOBD(ACCTLIB/INITSBS)
```

This command adds the job ACCTINIT as a job entry that starts automatically to the subsystem description ACCTINT in the library ACCTLIB. In this case, the job that starts automatically might be used to perform certain routines whenever the subsystem ACCTINT is started. When the subsystem is started, the job description INITSBS in ACCTLIB is used to obtain the attributes for this job and a job named ACCTINIT is automatically started in the subsystem.

ADDALRACNE (Add Alert Action Entry) Command



Notes:

P All parameters preceding this point can be specified in positional form.

1 A maximum of 5 repetitions

2 A maximum of 5 repetitions

Purpose

The Add Alert Action Entry (ADDALRACNE) command allows the user to add an action entry to the specified alert filter. This entry describes the actions that should be taken for an alert that has been assigned to the specified group. More information on alerts is in the *Alerts and DSNX Guide*.

Required Parameters

FILTER

Specifies the qualified name of the filter to which the action entry is added.

The name of the filter can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

filter-name: Specify the name of the filter.

GROUP

Specifies the group name to which the defined actions are to be applied. The group name is assigned from the selection criteria in the filter.

Optional Parameters

LOG

Specifies whether the alert is logged.

***NETATR:** The ALRLOGSTS network attribute controls the logging of this alert.

***YES:** The alert is logged.

***NO:** The alert is not logged.

ASNUSER

Specifies the user assigned to the alert.

***NONE:** No user is specified.

assigned-user: Specify a user name.

SEND

Specifies the destination to which the alert is to be sent. An alert cannot be sent to the local system or sent multiple times. The system checks for this action when the alert is sent.

***NONE:** The alert is not sent.

Element 1: Network Identifier

***FOCALPT:** Sends the alert to the system focal point. The focal point system is determined at send time.

***NETATR:** The LCLNETID value specified in the system network attributes is used.

network-ID: Specify the network ID of the destination node.

Element 2: Control Point Name

control-point-name: Specify the control point name of the destination system.

SNDDTAQ

Specifies the data queue in which an alert notification record is placed. Keyed data queues are supported.

***NONE:** No data queue is used.

The name of the data queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

Element 1: Data Queue Name

data-queue-name: Specify the name of the data queue.

Element 2: Data Queue Key

***NONE:** No key is used on the data queue.

data-queue-key: Specify the data queue key.

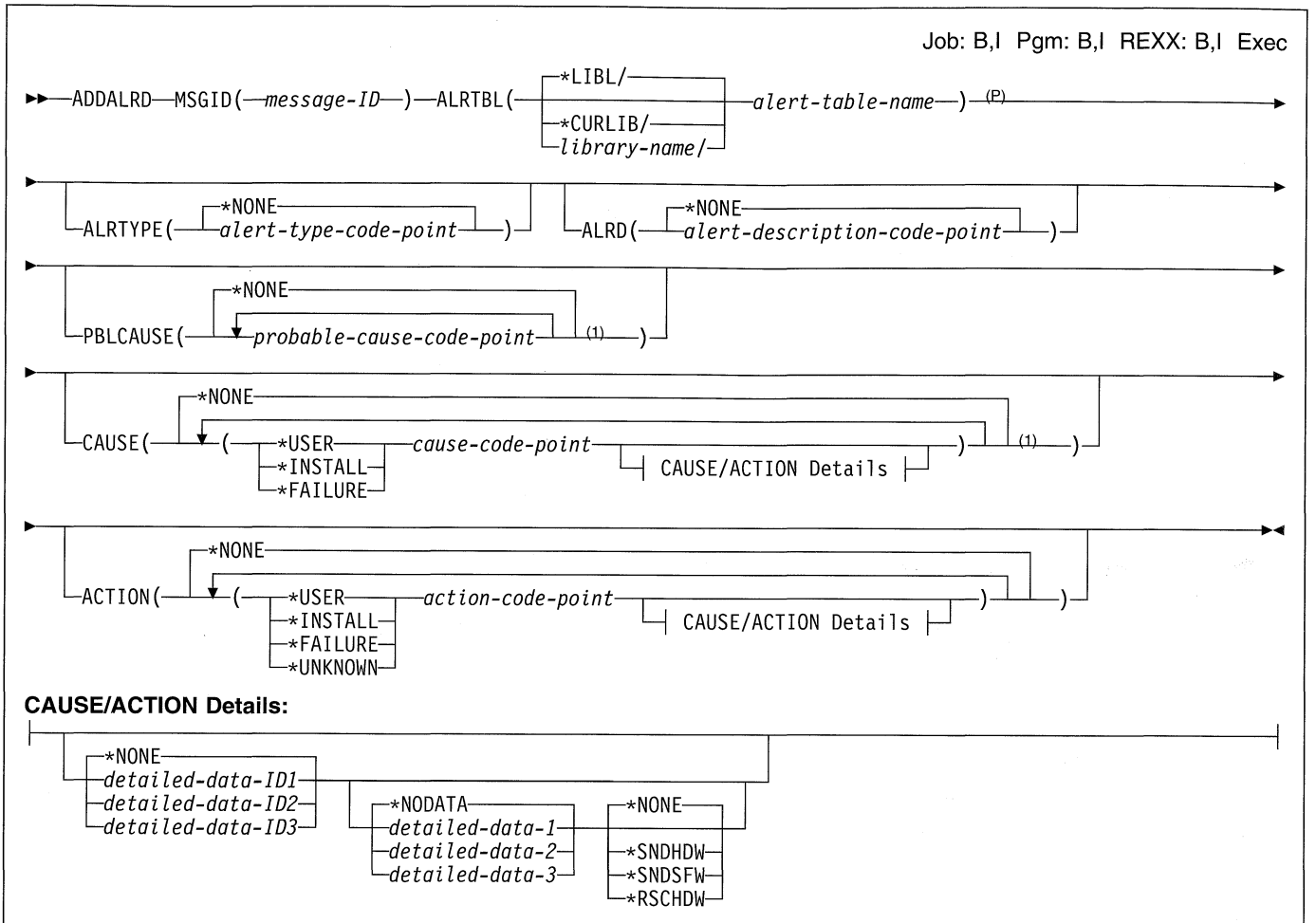
Example

```
ADDALRACNE  FILTER(MYLIB/MYFILTER)  GROUP(CHICAGO)
  LOG(*NETATR)  ASNUSER(CHICAGOOPR)
  SEND((*FOCALPT)(*NETATR.MILWKEE))
  SNDDTAQ(*LIBL/ALERTDTAQ)
```

This command defines the following actions for group CHICAGO:

1. Log the alert based on the ALRLOGSTS network attribute.
2. Send the alert to this system's focal point.
3. Send the alert to the system with control point name MILWKEE and a network id based on the LCLNETID value specified in the system network attributes.
4. Place an alert notification on data queue ALERTDTAQ.
5. Assign the alert to user CHICAGOOPR.

ADDALRD (Add Alert Description) Command



Notes:

1 A maximum of 99 repetitions

P All parameters preceding this point can be specified in positional form.

Purpose

The Add Alert Description (ADDALRD) command allows the user to create the description of an alert condition for a particular message identifier. The user provides the SNA generic alert code points that are used to create an alert. More information on alerts is in the *Alerts and DSNX Guide*.

Required Parameters

MSGID

Specifies the message identifier to which this alert description corresponds.

ALRTBL

Specifies the alert table in which this alert description is created.

The name of the alert table can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

alert-table-name: Specify the name of the alert table that is used.

Optional Parameters

ALRTYPE

Specifies the code point (a hexadecimal number) for the alert type.

***NONE:** There is no alert type code point for this alert description.

alert-type-code-point: Specify the alert type code point.

ALRD

Specifies the code point for the alert description.

***NONE:** There is no alert description code point for this alert description.

alert-description-code-point: Specify the alert description code point.

PBLCAUSE

Specifies probable causes, which are listed in order of decreasing probability. Up to 99 code points for probable cause can be listed.

***NONE:** There are no probable cause code points for this alert description.

probable-cause-code-point: Specify the probable cause code point.

CAUSE

Specifies user, install, or failure causes. Up to 99 causes can be specified.

***NONE:** There are no cause code points for this alert description.

Element 1: Type of Code Point

***USER:** A user cause code point follows.

***INSTALL:** An install cause code point follows.

***FAILURE:** A failure cause code point follows.

Element 2: Cause Code Point

cause-code-point: Specify the cause code point. Up to three detailed data qualifiers or one product identifier qualifier can be specified for each code point. A detailed data qualifier consists of a detailed data identifier code point and detailed data. Specify *NONE or *NODATA if there is no detailed data.

Element 3: First Detailed Data Identifier

***NONE:** There is no detailed data identifier code point for this cause.

detailed-data-ID: Specify the detailed data identifier code point used to identify the data. Detailed data identifiers can be specified up to three times in each session.

Element 4: Detailed Data for First Identifier

***NODATA:** There is no data for this cause.

detailed-data: Specify up to 40 characters of detail data. A substitution variable from the corresponding message description can be specified and the message data is substituted into the alert description when the alert is created.

Element 5: Product Identifier

***NONE:** There is no product identifier for this cause.

***SNDHDW:** Indicates the sender hardware, which is the AS/400 system.

***SNDSFW:** Indicates the sender software specified in the LICPGM keyword of the CRTALRTBL command.

***RSCHDW:** Indicates the failing resource hardware, which is determined by the resource hierarchy in the message description.

Note: The user can specify either 0 to 3 detailed data qualifiers or one product identifier qualifier, but not both.

ACTION

Specifies a recommended action for a user, install, or failure cause. Up to 99 recommended actions can be listed.

***NONE:** There are no recommended action code points for this alert description.

Element 1: Type of Action Code Point

***USER:** A user cause code point follows.

***INSTALL:** An install cause recommended action code point follows.

***FAILURE:** A failure cause recommended action code point follows.

***UNKNOWN:** A recommended action for a 'cause undetermined' error follows.

Element 2: Action Code Point

action-code-point: Specify the recommended action code point. Up to three detailed data qualifiers or one product identifier qualifier can be specified for each code point. A detailed data qualifier consists of a detailed data ID code point and detailed data. Specify *NONE *NODATA if there is no detailed data.

Element 3: First Detailed Data Identifier

***NONE:** There is no detailed data identifier code point for this action.

detailed-data-ID: Specify the detailed data identifier code point used to identify the data. Detailed data identifiers can be specified up to three times in each session.

Element 4: Detailed Data for First Identifier

***NODATA:** There is no data for this action.

detailed-data: Specify up to 40 characters of detail data. A substitution variable from the corresponding message description can be specified and the message data is substituted into the alert description when the alert is created.

Element 5: Product Identifier

***NONE:** There is no product identifier for this action.

***SNDHDW:** Indicates the sender hardware, which is the AS/400 system.

***SNDSFW:** The sender software, specified in the PRDID keyword of the CRTALRTBL command is used.

***RSCHDW:** Indicates that the failing resource hardware, which is determined by the resource hierarchy in the message description is used.

Note: The user can specify either 0 to 3 detailed data identifiers or one product identifier qualifier, but not both.

Example

```
ADDALRD   MSGID(USR1234)  ALRTBL(USER/USRMSGS)
  ALRTYPE(01) ALRD(3100)
  PBLCAUSE(1000 3121)
  CAUSE((*USER 6001)  (*FAILURE 1000)
  (*FAILURE 3121))  ACTION((*USER 1000)
  (*FAILURE 00B0 A5 'DSPMSG QSYSOPR')
  (*FAILURE F0A0 22 '&5')
  (*FAILURE 00E1 *NONE *NODATA  *NONE
  *NODATA  *NONE *NODATA  *SNDHDW))
```

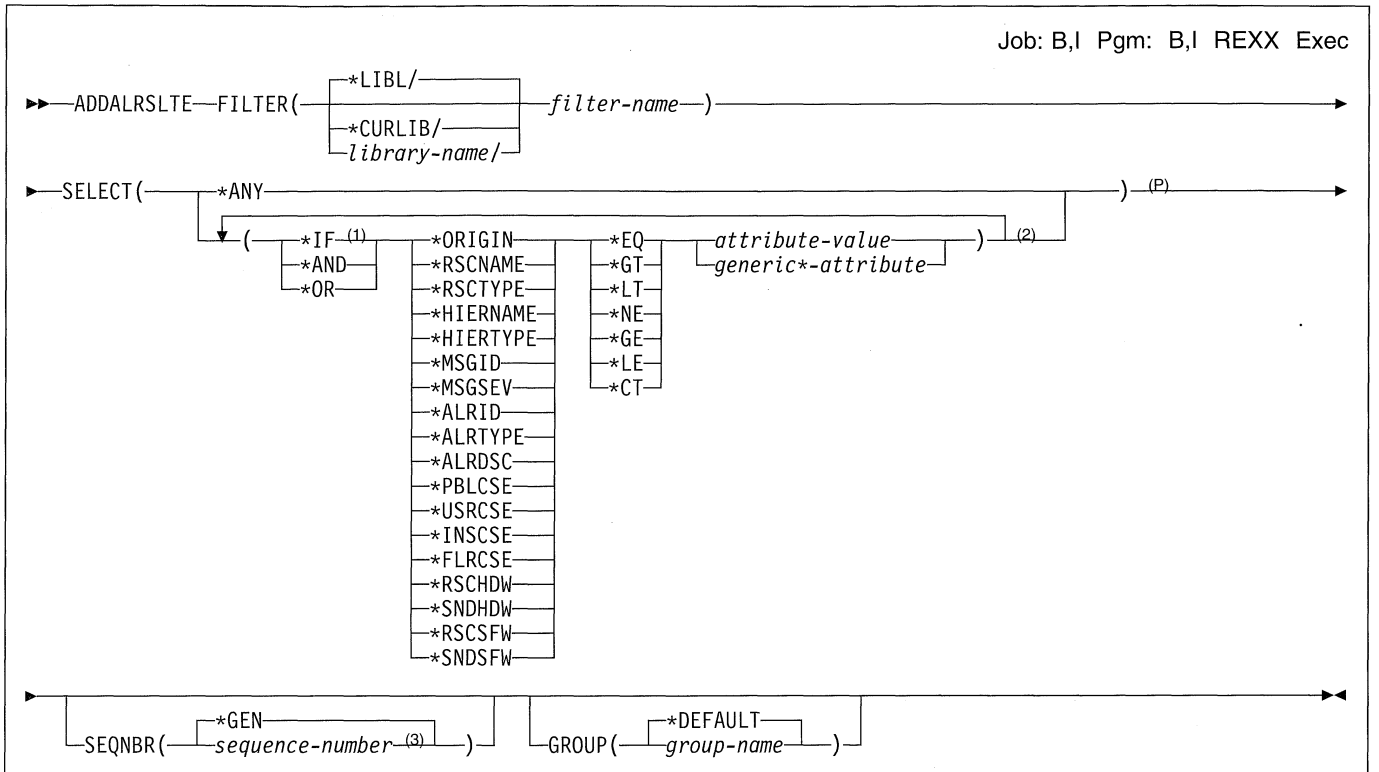
This command defines three recommended failure actions:

- '00B0', which requires a detailed qualifier. One detailed data qualifier is provided.
- The detailed data identifier code point is 'A5', which identifies the text 'Command', and
- The detailed data 'DSPMSG QSYSOPR'.

Failure recommended action 'F0A0' specifies a message substitution variable ('&5') as the detailed data. When the message 'USR1234' is sent, the message data for variable '&5' is put into the alert for the detailed data.

Failure action X'00E1' references a product identifier; in this case, it is the sending hardware AS/400. Place holders are needed for the detailed data qualifiers.

ADDALRSLTE (Add Alert Selection Entry) Command



Notes:

1 *IF must be specified as the first item only; either *AND or *OR must be specified when repeating selections.

2 A maximum of 10 repetitions

P All parameters preceding this point can be specified in positional form.

3 Range is 1 through 9999

Purpose

The Add Alert Selection Entry (ADDALRSLTE) command allows the user to add an alert selection entry to an alert filter. Selection entries are the criteria that categorize a group of alerts. More information on alerts is in the *Alerts and DSNX Guide*.

Required Parameters

FILTER

Specifies the qualified name of the filter being added.

The name of the filter can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

filter-name: Specify the name of the filter.

SELECT

Specifies the comparisons to be made to determine if the alert belongs in the specified group. The selection entry results in a successful match with an alert when the data in the alert satisfies the relationships specified on the SELECT parameter. Up to 10 attribute values can be compared to the alert.

***ANY:** Any alert matches this selection record. Specify the conditions under which an alert matches the selection entry. Each condition must contain the following four elements:

1. One of the logical operators *IF, *AND, or *OR
2. The attribute compared
3. One of the relational operators
4. The attribute value

Element 1: Logical Operator

***IF:** Identifies the first condition that must be satisfied.

***AND:** The conditions on both sides of the *AND must be satisfied.

***OR:** One of the conditions on each side of the *OR must be satisfied.

If there is one set or several sets of conditions, the *IF value must be specified as the first value in the first set of comparison values. If more than one set of conditions are specified, *AND or *OR must be specified as the first value in each set after the first. Each condition must be enclosed in parentheses. *AND is evaluated before *OR.

Element 2: Attribute

***ORIGIN:** Specifies whether the alert is generated or received. The valid values for this attribute are L (Locally generated) or R (Received).

***RSCNAME:** Specifies the name of the failing resource. The value for this attribute must be a 8-character name.

***RSCTYPE:** Specifies the type of the failing resource. The value for this attribute must be a 3-character resource type (for example, TAP or DKT).

***HIERNAME:** Specifies all of the resources in the alert resource hierarchy. The alert resource hierarchy is the list of resources, separated by blanks, displayed on the Work with Alerts (WRKALR) command detailed data displays. The value for this attribute can be a list of up to 5 resource names separated by a blank, unless the value is used with the *CT relational operator. If the *CT value is used, the selection relation can test to see if the given resource name is found anywhere within the hierarchy. This attribute contains the resource names from the hierarchy only.

***HIERTYPE:** Specifies all of the resource types in the alert resource hierarchy. The resource types match the resource names specified on the *HIERNAME attribute. The value for this attribute can be a list of up to 5 resource types (1 to 3 characters in length) separated by a blank, unless the value is used with the *CT relational operator. If the *CT value is used, the selection relation can test to see if the given resource type is found anywhere within the hierarchy.

***MSGID:** Specifies the message identifier.

***MSGSEV:** Specifies the message severity. This value must be greater than or equal to -2,147,483,647 and less than or equal to 2,147,483,647.

***ALRID:** Specifies the alert identifier. The alert identifier is displayed on the Work with Alerts (WRKALR) command detailed data display. The value for this attribute must be an 8-digit hexadecimal number unless it is used with the *CT relational operator. If the *CT operator or a wildcard character is used, the attribute must have an even number of digits up to a maximum of 8. The alert ID may not be a valid comparison for AS/400 alerts created after problem analysis.

***ALRTYPE:** Specifies the alert type code point that is in the alert. The value for this attribute is a 2 digit hexadecimal number.

***ALRDSC:** Specifies the alert description code point that is in the alert. The value for this attribute must be an 4-digit hexadecimal number unless it is used with the *CT relational operator. If the *CT operator or a wildcard character is used, the attribute must have an even number of digits up to a maximum of 4.

***PBLCSE:** Specifies the probable cause code point that is in the alert. The value for this attribute must be an 4-digit hexadecimal number unless it is used with the *CT relational operator. If the *CT operator or a wildcard character is used, the attribute must have an even number of digits up to a maximum of 4.

***USRCSE:** Specifies the first user cause code point that is in the alert. The value for this attribute must be an 4-digit hexadecimal number unless it is used with the *CT relational operator. If the *CT operator or a wildcard character is used, the attribute must have an even number of digits up to a maximum of 4.

***INSCSE:** Specifies the first install cause code point that is in the alert. The value for this attribute must be an 4-digit hexadecimal number unless it is used with the *CT relational operator. If the *CT operator or a wildcard character is used, the attribute must have an even number of digits up to a maximum of 4.

***FLRCSE:** Specifies the first failure cause code point that is in the alert. The value for this attribute must be an 4-digit hexadecimal number unless it is used with the *CT relational operator. If the *CT operator or a wildcard character is used, the attribute must have an even number of digits up to a maximum of 4.

***RSCHDW:** Specifies the failing hardware resource information in the alert. This information is displayed on the Work with Alerts (WRKALR) command detailed data displays. Specify a value for this attribute using the following form:

```
'tttt mmm ss-ssssssss'
'tttt mmm ss-sssss'
'tttt mmm sssssss'
'tttt mmm sssss'
```

where `tttt` is the machine type, `mmm` is the model number, and `sssssssss` is the serial number. Use this format to match a particular hardware resource or use a part of the hardware value with the *CT relational operator to provide a partial match.

***SNDHDW:** Specifies the sending hardware resource information in the alert. This information is displayed on the Work with Alerts (WRKALR) command detailed data displays. Specify a value for this attribute using the following form:

```
'tttt mmm ss-ssssssss'
'tttt mmm ss-sssss'
'tttt mmm sssssss'
'tttt mmm sssss'
```

where `tttt` is the machine type, `mmm` is the model number, and `sssssssss` is the serial number. Use this format to match a particular hardware resource or use a

part of the hardware value with the *CT relational operator to provide a partial match.

***RSCSFW:** Specifies the failing software resource information in the alert. This information is displayed on the Work with Alerts (WRKALR) command detailed data displays. Specify a value for this attribute using the following form:

```
'ppppppp vv rr mm'
```

where ppppppp is the licensed program identifier, vv is the version number, rr is the release number, and mm is the modification level. Use this format to match a particular software resource or use a part of the software value with the *CT relational operator to provide a partial match.

***SNDSFW:** Specifies the sending software resource information in the alert. This information is displayed on the Work with Alerts (WRKALR) command detailed data displays. Specify a value for this attribute using the following form:

```
'ppppppp vv rr mm'
```

where ppppppp is the licensed program identifier, vv is the version number, rr is the release number, and mm is the modification level. Use this format to match a particular software resource or use a part of the software value with the *CT relational operator to provide a partial match.

Element 3: Relational Operator

***EQ:** The attribute in element 2 must be equal to the value specified in element 4.

***GT:** The attribute in element 2 must be greater than the value specified in element 4.

***LT:** The attribute in element 2 must be less than the value specified in element 4.

***NE:** The attribute in element 2 must not be equal to the value specified in element 4.

***GE:** The attribute in element 2 must be greater than or equal to the value specified in element 4.

***LE:** The attribute in element 2 must be less than or equal to the value specified in element 4.

***CT:** The attribute in element 2 must contain the value specified in element 4.

Element 4: Attribute Value

attribute-value: Specify the value (a maximum of 60 characters) to be compared with the contents of the specified attribute. The value must be specified in apostrophes if it contains blanks or special characters and must be in character format. If a CL variable is specified for the value, it must be a character variable.

generic-attribute-value:* Specify the generic attribute value. Generic attribute values are only allowed with the *EQ and *NE operator. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all attributes with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete attribute name. If the complete attribute name is specified, and multiple libraries are searched, multiple attributes can be added only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

Optional Parameters

SEQNBR

Specifies the sequence number of the alert selection entry. Selection entries are evaluated in order by sequence number.

***GEN:** Allows the system to generate the sequence number. The sequence number will be greater than all previous selection entries.

sequence-number: Specify a number from 1 through 9999.

GROUP

Specifies the group that an alert is assigned to if the alert matches the criteria specified on the SELECT parameter.

***DEFAULT:** The alert is assigned to the *DEFAULT group. The *DEFAULT group is automatically added when a filter is created.

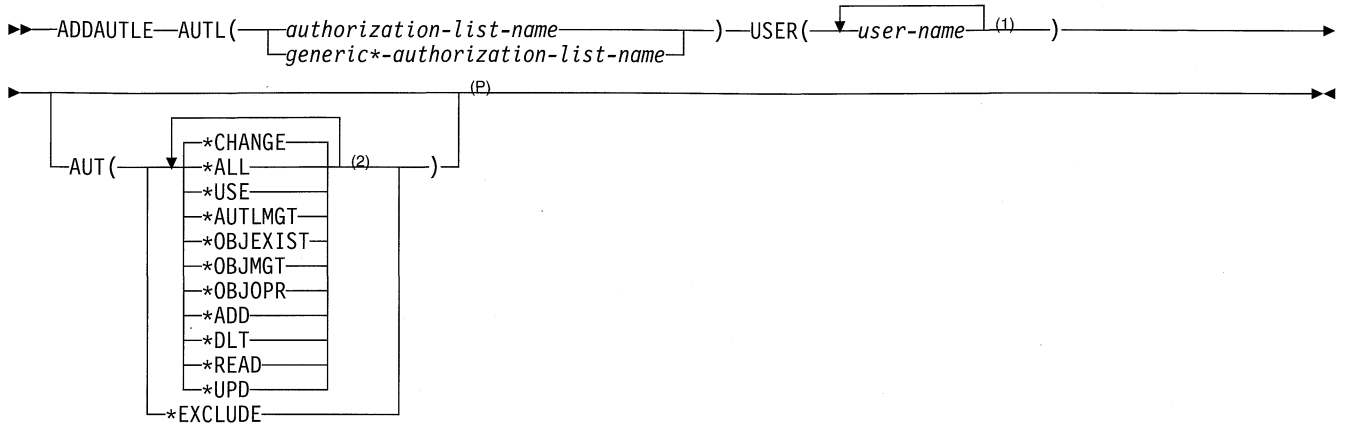
group-name: Specify a group name to which the alert is assigned.

Example

```
ADDALRSLTE  FILTER(MYLIB/MYFILTER)
  SELECT((*IF *RSCNAME *EQ CHICAGO1)
  (*AND *RSCTYPE *EQ CP))
  SEQNBR(*GEN)  GROUP(CHICAGO)
```

This command adds selection entry 0010 to the filter MYFILTER in library MYLIB (a 0010 is generated because no entries have been added to the filter). Any alerts that have a resource name of 'CHICAGO1' and a resource type of 'CP' (control point) are assigned to group CHICAGO.

ADDAUTLE (Add Authorization List Entry) Command



Notes:

1 A maximum of 50 repetitions

2 Select one or more, 8 maximum

P All parameters preceding this point can be specified in positional form.

Purpose

The Add Authorization List Entry (ADDAUTLE) command allows the user to add entries to an authorization list. An entry consists of a user's name and the authorities associated with that user on the authorization list. Both the authorization list and the user profile must exist. If the specified user is already on the list, a message is issued and the user's authorities on the list are not changed.

The users who can use this command to add users to an authorization list are: the owner of the authorization list, a user with authorization list management (AUTLMGT) authority on the authorization list, or a user with all object (ALLOBJ) authority.

When the ADDAUTLE command is used to add a user to an authorization list, the user must specify the name of the authorization list, a list of authorized users, and a list of authorities specified for the list. Each user on the list is given the authorities specified on the command.

Restrictions:

1. Authorization list management authority allows a user to manage the authorization list and, therefore, to manage the authorities for all objects secured by the list.
2. Only the owner of the list or a user with *ALLOBJ authority can add a user with *AUTLMGT authority.
3. A user with *AUTLMGT authority can add users and give specific authorities only to the *AUTLMGT level.

Required Parameters

AUTL

Specifies the name or generic name of the authorization list to which the users are being added. The authorization list must already exist.

authorization-list-name: Specify the name of the authorization list to which the user profile name is added.

generic-authorization-list-name:* Specify the generic name of the authorization list to which the user profile names are being added. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be added only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

USER

Specifies a list of user names to be added to the authorization list. Up to 50 user names can be specified. If a user ID is already on the authorization list, a message is issued and the user's authorities are not changed.

Optional Parameter

AUT

Specifies the authority given to users specified on the USER parameter. Users must have *AUTLMGT authority to manage the authorization list.

***CHANGE:** The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority. If the user profile name is an authorization list, the user cannot add, change, or remove user IDs.

***ALL:** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence and specify the security for the object, change the object, and perform basic functions on the object. The user can change ownership of the object.

***USE:** The user can perform basic operations on the object, such as running a program or reading a file. The user cannot change the object. *USE authority provides object operational authority and read authority.

***EXCLUDE:** The user cannot access the object.

***AUTLMGT:** Authorization list management authority provides the authority to add users to the authorization list, to change users' authorities on the authorization list, or to remove users from the authorization list, to rename an authorization list, or to create a duplicate authorization list.

***OBJEXIST:** Object existence authority provides the authority to control an object's existence and ownership. These authorities are necessary for users who want to delete an object, free storage for an object, perform save and restore operations for an object, or transfer ownership of an object. A user with special save system (*SAVSYS) authority does not need object existence authority to save or restore objects. Object existence authority is required to create an object that has been named by an authority holder.

***OBJMGT:** Object management authority provides the authority to specify the security for an object, to move or rename an object, and to add members to database files.

***OBJOPR:** Object operational authority provides authority to look at the description of an object and to use the object as determined by the data authorities held by the user.

***ADD:** Gives the authority to add entries to an object (for example, job entries to a queue or records to a file).

***DLT:** Delete authority allows the user to remove entries from an object, for example, remove messages from a message queue or records from a file.

***READ:** Read authority provides the authority needed to show the contents of an entry in an object or to run a program.

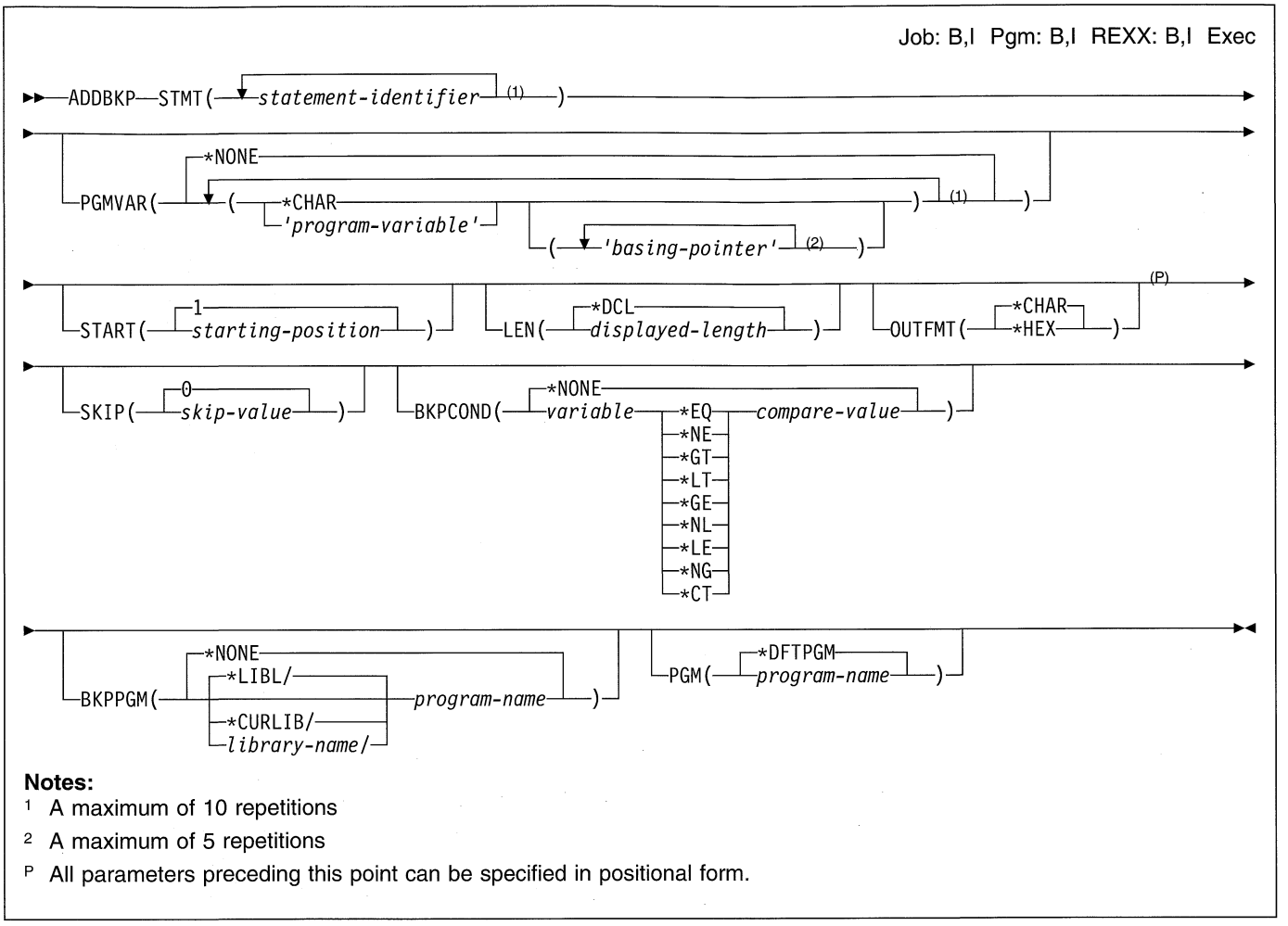
***UPD:** Update authority provides the authority needed to change the entries in an object.

Example

```
ADDAUTLE  AUTL(PAYROLL)  USER(TOM)
  AUT(*ALL *AUTLMGT)
```

This command adds user TOM to the PAYROLL authorization list and gives him all authority to the objects secured by the authorization list. TOM also has authority to manage the authorization list.

ADDBKP (Add Breakpoint) Command



Notes:

1 A maximum of 10 repetitions

2 A maximum of 5 repetitions

P All parameters preceding this point can be specified in positional form.

Purpose

The Add Breakpoint (ADDBKP) command sets up to 10 breakpoints in a program.

A breakpoint is a location in a program where processing stops and control is given to the user or to a specified program. The breakpoint is set when a statement number or label of a command or machine instruction is specified. The program is stopped just before processing begins on the statement (or machine instruction) on which the breakpoint is set.

This command shows the values of certain program variables when a breakpoint in the program is reached. As many as 10 variables per breakpoint can be specified, and as many as 10 breakpoints per command can be set. However, the same program variables apply to every breakpoint specified in the command. Different ADDBKP commands must be used to specify different sets of variables for each breakpoint.

This command specifies conditional breakpoints in which the program is stopped when a condition is true. This condition involves two program variables or one program variable and a constant. When using conditional breakpoints, it is possible to stop the program when a program variable becomes a certain value.

A conditional breakpoint can also be specified by specifying a skip value. The program does not stop until the breakpoint statements have been processed as many times as the skip number indicates. After that, the breakpoint causes the program to stop.

When a breakpoint is reached in the interactive debugging environment, a display is shown to the user that identifies which breakpoint has been reached and (optionally) the values of the specified program variables when the program is stopped. A message is also written to the job log when the breakpoint is reached. From the display, the user presses F10 to show the command entry display, presses F3 to exit the display and cancel the program, or presses the Enter key to allow the program to continue running.

When a breakpoint is reached in the batch debugging environment, the breakpoint information is written to a printer file and, optionally, another program is called to take action on

the breakpoint condition. The name of the called program is specified in the BKPPGM parameter.

When an interactive job is debugging another job, and a breakpoint is reached in the debugged job, a breakpoint display is shown. This display appears in the debugging job, interrupting what was previously being shown. The user must press the Enter key, allowing the stopped program to continue, before returning to the previous display.

Restrictions:

1. This command is valid only in the debug mode. For more information on how to start the debug mode, refer to the description of the STRDBG (Start Debug) command.
2. This command cannot be used if the user is servicing another job, and that job is on a job queue, or is being held, suspended, or ended.
3. This command cannot be used to add breakpoints to a bound program.

Required Parameter

STMT

Specifies the statement identifiers of up to 10 statements or machine instructions in the program at which breakpoints are set. The program stops before processing a statement specified as a breakpoint.

The list can contain up to 10 identifiers (statement numbers, program labels, or machine instruction numbers) that are valid for the program specified by the PGM parameter. At least one identifier is needed. If a machine instruction number is specified, a slash must be placed in front of the number and both the slash and the number must be enclosed in apostrophes; for example, `STMT('/21')`.

In high-level language programs, different statements and/or labels can be mapped to the same internal instruction. This happens when several statements that do not operate on variables directly (such as DO, END, and comments) follow one another in a program. The intermediate representation of a program list is used to determine which statements can be mapped to the same instruction.

Because different statements can be mapped to the same instruction, adding a breakpoint can redefine a previous breakpoint that was added for a different statement. When this occurs, the new breakpoint replaces the previously defined breakpoint.

Optional Parameters

PGMVAR

Specifies the names of up to 10 program variables shown that are in a high-level language or a machine instruction program. The name and value of each program variable is shown when any of the breakpoints specified in the STMT parameter are reached. During a run, the program stops *before* processing a statement specified as a breakpoint.

Note: In some high-level languages such as RPG, variables that are declared but not referred to in the program cannot be specified on the PGMVAR parameter.

***NONE:** No program variables are shown for any of the breakpoints specified.

Element 1: Program Variables

***CHAR:** This special value is specified instead of a variable name if a basing pointer is also specified. This special value displays a character view of a pointer to be shown without the use of a based variable.

'program-variable': Specify the names of up to 10 program variables, separated by blanks, shown when a breakpoint is reached. The names must be enclosed in apostrophes if they contain special characters. For example, a CL variable, &VAR, must be specified as `PGMVAR('&VAR')`.

If the program variable is an array, the subscripts representing an element in the array can be specified as a breakpoint. If an array name is specified without any subscripts, all of the array elements are recorded. A single-dimensional cross-section can also be specified. Up to 132 characters can be specified for this program variable entry. This includes any qualifiers, subscripts, embedded blanks, parentheses, and commas. It does not include the enclosing apostrophes when special characters are used.

An integer, a machine-interface object-definition-table-vector (MI ODV) number, asterisk (single-dimensional cross-section), or a numeric variable name can be specified for a subscript. For more information on testing and debugging at machine interface level and on the program-variable value, refer to the *CL Programmer's Guide* and Appendix C, "Parameter Values Used for Testing and Debugging." Some examples follow:

```
PGMVAR(A)
PGMVAR('A(2,B)')
PGMVAR('B(I1,*,I3)')
PGMVAR('VAR1 OF A(I,J IN B)')
```

Element 2: Basing Pointers

'basing-pointer': Specify up to five basing pointers for the program variable being shown. In some languages, the program variable can be based on a pointer variable. Each basing pointer name must be enclosed in apostrophes if it contains special characters.

If the basing-pointer is an array, the subscripts representing an element in the array must be specified. Up to 132 characters can be specified for a basing-pointer name. This includes any qualification, subscripts, embedded blanks, parentheses, and commas. It does not include the enclosing apostrophes when special characters are used. An integer, an MI ODV number, or a numeric variable name can be specified for a sub-

script. For more information on the basing-pointer value, refer to Appendix C, "Parameter Values Used for Testing and Debugging." Some examples are:

```
PGMVAR(('VAR1(B,5)' 'PTR2(C,P2)'))
PGMVAR((VAR2 (BASEPTRA BASEPTRB)))
```

START

Specifies, for string variables only, the starting position in the string from which its value is shown when the breakpoint is reached. If more than one string variable is specified in the PGMVAR parameter, the same starting position value is used for each one. For a bit string, the value specifies the starting bit position; for a character string, the value specifies the starting character position.

For conditional breakpoints, the START parameter also specifies the start in the string where the comparison is made.

1: The variable is shown from the first position through the length specified on the LEN parameter.

starting-position: Specify the first position of the program variable being shown.

The START value specified must not be larger than the maximum string length for any variable specified, except that START(1) is allowed if the maximum length for a string is 0. The LEN value, plus the START position minus one, must not be greater than the maximum string length. These checks are made for each string variable specified in the PGMVAR parameter.

LEN

Specifies, for string variables only, the length of the string shown when the breakpoint is reached, starting at the position specified by the START parameter. If more than one string variable is specified in the PGMVAR parameter, the same value is used for each one. For a bit string, the value specifies the number of bits shown; for a character string, the value specifies the number of characters shown.

For conditional breakpoints, the LEN parameter also specifies the length of the string where the comparison is made.

***DCL:** The string variable is shown to the end of the string or for a value of 200 bytes, whichever is less. If the string variable has a maximum length of 0, only LEN(*DCL) is allowed.

displayed-length: Specify the length of the data shown. The length (as well as the combination of START and LEN) must be no greater than the length of the shortest string specified in the PGMVAR parameter.

OUTFMT

Specifies the format in which the objects are shown.

***CHAR:** Variables are shown in character form.

***HEX:** Variables are shown in both character format and hexadecimal format.

SKIP

Specifies the number of times the statement or statements on the STMT parameter must be processed before the program is stopped.

0: The program stops immediately when the statement or statements on the STMT parameter are processed. No skipping of breakpoints is done.

skip-value: Specify the number of times the statements on the STMT parameter must be processed before the program is stopped. If there is more than one statement specified, each statement will have its own independent skip value. There is a separate skip count for each statement.

BKPCOND

Specifies a defined condition that must be true before the program is stopped. The condition is tested before any statement on the STMT parameter is processed. If the condition is false, the breakpoint does not stop the program. If the condition is true, the program is stopped.

***NONE:** No breakpoint condition is specified.

Element 1: Variable

variable: Specify a variable to be used in the breakpoint condition. *PGMVAR1 indicates the first variable, *PGMVAR2 the second, and so on. Only numeric, character, or bit variables can be specified.

Element 2: Operator

operator: Specify the type of comparison to be done for a conditional breakpoint. The following comparisons are allowed:

```
*EQ - equal to
*NE - not equal to
*GT - greater than
*LT - less than
*GE - greater than or equal to
*NL - not less than (same as *GE)
*LE - less than or equal to
*NG - not greater than (same as *LE)
*CT - contains
```

The *CT operator compares character strings to determine whether one character string contains one or more occurrences of another character string. This comparison is for an exact match, and it is case sensitive.

Element 3: Comparing a Constant or Variable

compare-value: Specify a constant or another variable to compare with the variable. If a constant is specified, it must be the same type as the variable. If the variable is numeric, the constant must be a number. If the variable is a bit, the constant must be a string containing only 1's and 0's. If the variable is a character, the compare value is treated as a character string, even if a number is specified.

If another program variable is specified, it is compared with the variable. The variables must be of the same type. If the variables are numeric, they must both be

floating point or not floating point. For example, a packed number cannot be compared with a floating point number.

When comparing two non-floating point variables, or a non-floating point variable and a constant, the total number of digits needed to represent them must not exceed 31. For example, a PACKED(24,2) and a PACKED(24,20) cannot be compared. The first variable requires 22 digits to the left of the decimal point and two digits to the right. The second variable requires four digits to the left of the decimal point and 20 digits to the right. To compare these variables would require a variable with 22 digits to the left of the decimal point and 20 digits to the right. This exceeds the maximum number of allowed digits, 31.

When comparing two character strings, the shorter of the two is padded with blanks. When comparing two bit strings, they must both be of the same length.

The SKIP and BKPCOND parameters can be used together. In this case, the breakpoint condition is not evaluated until the breakpoint has skipped the number of times specified by SKIP. After that, the breakpoint condition is evaluated and the program stops if the condition is true.

BKPPGM

Specifies the qualified name of the user-supplied program (if any) to call when a breakpoint is reached in the program specified by the PGM parameter. When the program specified on the BKPPGM parameter is called, it is passed four parameters that identify: the program name, the recursion level, the high-level language statement identifier, and the machine instruction number at which the breakpoint occurred. Those four parameters have the following formats:

1. Program name (10 bytes). The name of the program in which the breakpoint was reached.
2. Recursion level (5 bytes). The recursion level number of the program in which the breakpoint was reached. This value is a 1- to 5-digit number that is padded on the right with blanks.
3. Statement identifier (10 bytes). The high-level language program statement identifier that was reached. This statement identifier is the statement identifier specified in the Add Breakpoint (ADDBKP) command that defined the breakpoint. If a machine instruction number was used to specify the breakpoint, this parameter contains a slash (/) followed by a 4-digit hexadecimal machine instruction number.
4. Instruction number (5 bytes). The machine instruction number that corresponds to the high-level language statement at which the breakpoint was reached. No slash appears in front of this machine instruction number. It consists of 1 to 4 hexadecimal characters that represent the MI instruction number, followed by one or more blanks. If a machine instruction number is passed in the third parameter, the numbers in the third and fourth parameters are the same.

All the parameter values are left-adjusted and padded with blanks. When the called program returns, the program being debugged continues processing, with the statement with the breakpoint on it.

***NONE:** No breakpoint-handling program is called when any breakpoint specified in this ADDBKP command is reached in the batch environment. The interrupted program continues processing.

The name of the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the user-supplied program to call if any of the breakpoints on this command are reached while debugging in a batch environment. The program specified should not be the same as the program specified in the PGM parameter. If they are the same, the results are unpredictable. After the called program runs, it returns control to the interrupted program, which continues processing.

PGM

Specifies the name of the program to which the breakpoints are added.

***DFTPGM:** The breakpoints are added to the program currently specified as the default program in debug mode.

program-name: Specify the name of the program to which the breakpoints are added. The program must already be in debug mode.

Examples

Example 1: Adding Breakpoints in Debug Mode

```
ADDBKP  STMT(150 RTN1 205)
  PGMVAR('&TEMP' '&INREC')
```

This command establishes breakpoints at CL statement numbers 150 and 205 and at the label RTN1 for the default program in debug mode. When any of these breakpoints is reached, the CL variables &TEMP and &INREC are automatically shown. Note that the CL variables must include a leading ampersand (&) and be specified within apostrophes.

Example 2: Adding Breakpoints to HLL Program

```
ADDBKP  STMT(100)  PGMVAR('AMOUNT(200)')
  PGM(MYPROG)
```

Assume in this example that MYPROG is a high-level language program being debugged in an interactive environment and that the program variable AMOUNT is a 250-element array in MYPROG. This command adds a breakpoint to statement 100 in MYPROG. When MYPROG is started, the program stops processing at statement 100, and the value of the 200th element of the AMOUNT array is shown. If AMOUNT had been specified without a subscript, all of the array elements would have been shown.

Example 3: Program Stops After Processing Statement 10 Times

```
ADDBKP   STMT(10)  SKIP(1000)
```

This command causes the default program to stop when statement 10 is processed 1000 times (the breakpoint is skipped 1000 times).

Example 4: Program Stops After Processing Multiple Statements

```
ADDBKP   STMT(10 20 30)  SKIP(50)
```

This command causes the default program to stop when statements 10, 20, and 30 are processed 50 times.

Example 5: Conditional Breakpoint

```
ADDBKP   STMT(10)  PGMVAR(X)  BKPCOND(PGMVAR1 *EQ 5)
```

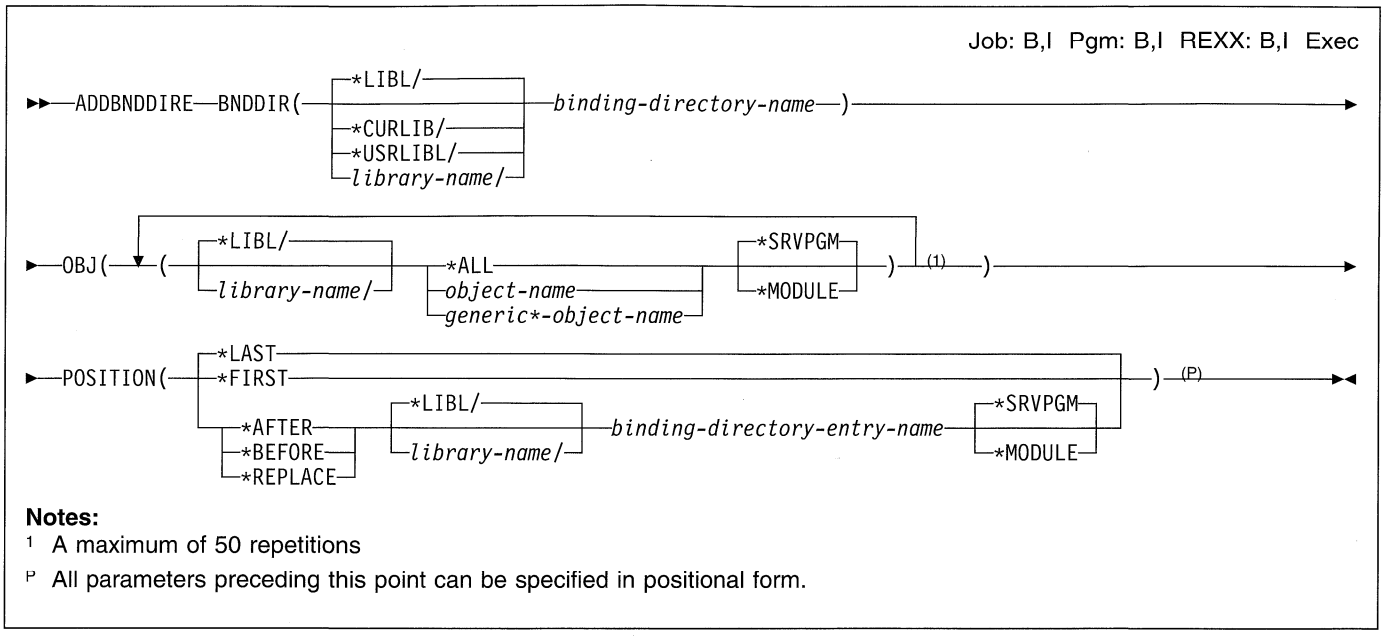
This command stops the default program at statement 10 when variable X is equal to five.

Example 6: Conditional Breakpoint

```
ADDBKP   STMT(20)  PGMVAR((S1) (S2))  SKIP(100)
  BKPCOND(*PGMVAR1 *CT *PGMVAR2)
```

This command stops after statement 20 has been processed 100 times, and then only if the character string S2 occurs in the character string S1.

ADDBNDDIRE (Add Binding Directory Entry) Command



Notes:

1 A maximum of 50 repetitions

P All parameters preceding this point can be specified in positional form.

Purpose

The Add Binding Directory Entry (ADDBNDDIRE) command adds an entry to the binding directory.

Restrictions:

1. You must have *USE authority for the library where the binding directory is being updated.
2. You must have object operational and *ADD authority to the binding directory.
3. You must have *READ authority to the specified library when using generic processing.

Required Parameters

BNDDIR

Specifies the binding directory to which an entry is added.

The name of the binding directory can be qualified by one of the following library values:

*LIBL: All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

***USRLIBL:** Only the libraries in the user portion of the job's library list are searched.

library-name: Specify the name of the library to be searched.

binding-directory-name: Specify the name of the binding directory to be updated.

OBJ

Specifies the object name to be added to the binding directory.

Element 1: Name of the Object to be Added

The name of the binding directory can be qualified by one of the following library values:

*LIBL: All libraries in the user and system portions of the job's library list are searched.

library-name: Specify the name of the library to be searched.

***ALL:** All objects of the specified type residing in the specified library are added.

object-name: Specify the object to be added.

generic-object-name:* Specify the generic name of the object. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be added only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

Element 2: Type of Object to be Added

*SRVPGM: Indicates the object to be added is a service program.

***MODULE:** Indicates the object to be added is a module.

POSITION

Specifies the position in the binding directory where the list of objects is added.

***LAST:** The list of objects is added to the end of the binding directory entries.

***FIRST:** The list of objects is inserted prior to the first binding directory entry.

Element 1: Object Position

***AFTER:** The list of objects is added to the binding directory after the binding directory entry specified on this parameter. The entry specified must currently exist in the binding directory.

***BEFORE:** The list of objects is added to the binding directory before the binding directory specified on this parameter. The entry specified must currently exist in the binding directory.

***REPLACE:** The object specified on the OBJ parameter replaces the binding directory entry specified on this parameter. The entry specified must currently exist in the binding directory. Only one entry can be specified on the OBJ parameter.

Element 2: Binding Directory Entry Name

The name of the binding directory can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

library-name: Specify the name of the library to be searched.

binding-directory-entry-name: Specify a binding directory entry name that exists in the specified binding directory.

Element 3: Binding Directory Entry Type

***SRVPGM:** Indicates the directory entry is a service program.

***MODULE:** Indicates the directory entry is a module.

Examples

Example 1:

```
ADDBNDDIRE   BNDDIR(TESTBNDDIR)
  OBJ((TESTOBJ))  POSITION(*LAST)
```

This command adds a binding directory entry for service program TESTOBJ in library *LIBL to the end of the list of binding directory entries found in the binding directory TESTBNDDIR.

Example 2:

```
ADDBNDDIRE   BNDDIR(TESTBNDDIR)
  OBJ((TESTLIB/TESTOBJ *MODULE))  POSITION(*FIRST)
```

This command adds a binding directory entry for module TESTOBJ in library TESLIB to the beginning of the binding directory entries found in the binding directory TESTBNDDIR.

Example 3:

```
ADDBNDDIRE   BNDDIR(TESTBNDDIR)
  OBJ((TESTLIB/TESTOBJ *MODULE)  (TESTOBJ2))
  POSITION(*FIRST)
```

This command adds a binding directory entry for module TESTOBJ in library TESTLIB followed by an entry for service program TESTOBJ2 in the library list to the beginning of the binding directory entries found in the binding directory TESTBNDDIR.

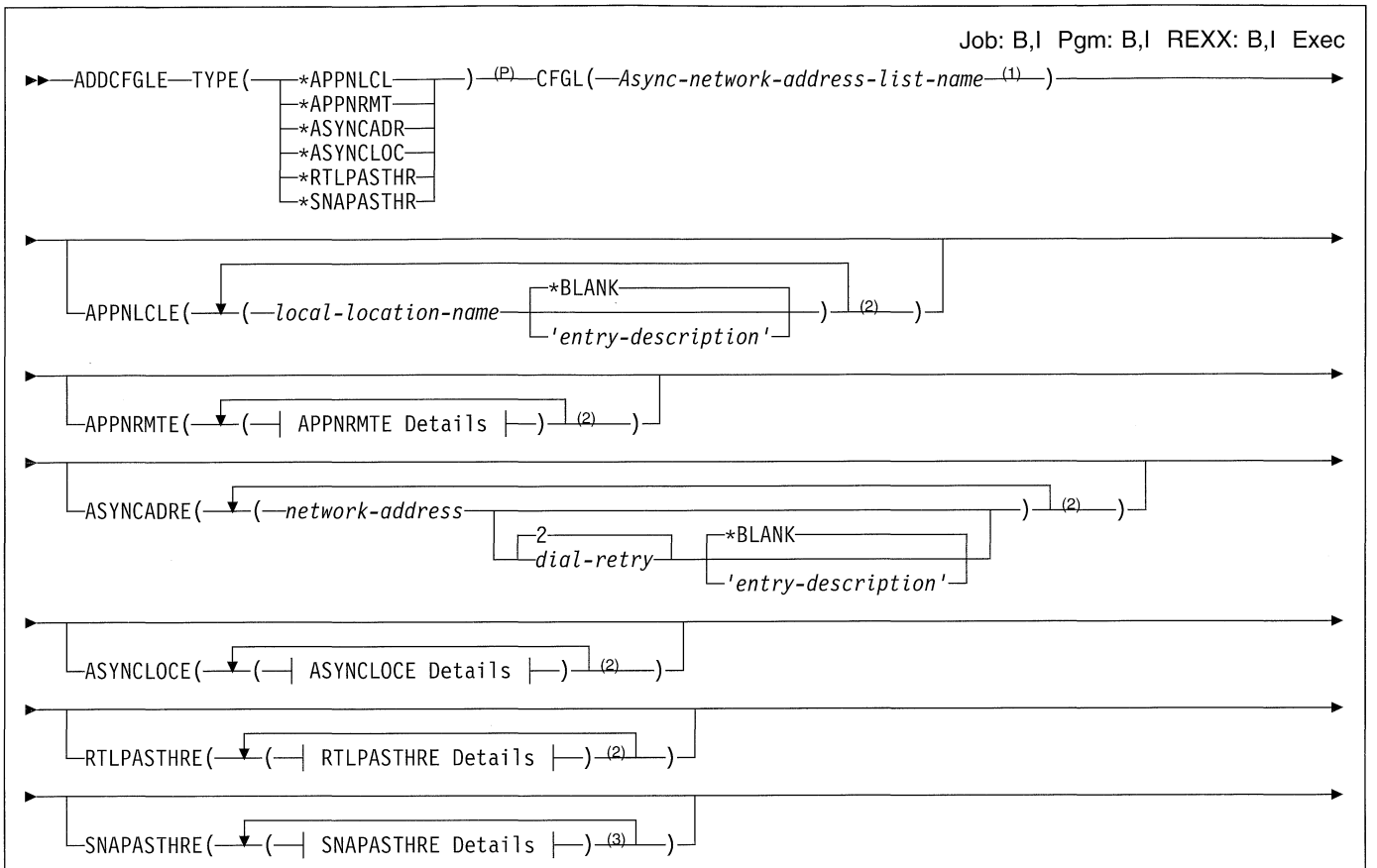
Example 4:

```
ADDBNDDIRE   BNDDIR(TESTBNDDIR)
  OBJ((TESTLIB/TESTOBJ *MODULE) (TESTOBJ2))
  POSITION(*BEFORE  TESTMOD *MODULE)
```

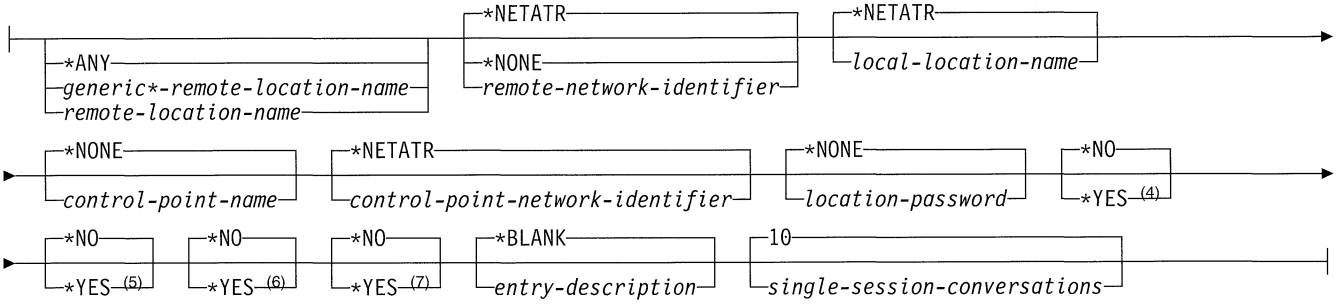
This command adds a binding directory entry for module TESTOBJ in library TESTLIB, followed by an entry for service program TESTOBJ2 in library *LIBL prior to the binding directory entry for module TESTMOD in library *LIBL found in the binding directory TESTBNDDIR.

The binding directory entry for module TESTMOD in library *LIBL must be found in the binding directory TESTBNDDIR for this operation to be successful.

ADDCFGLE (Add Configuration List Entries) Command



APPNRMTE Details:



Notes:

1 Required only for TYPE(*ASYNCADR)

2 A maximum of 50 repetitions

3 A maximum of 254 repetitions

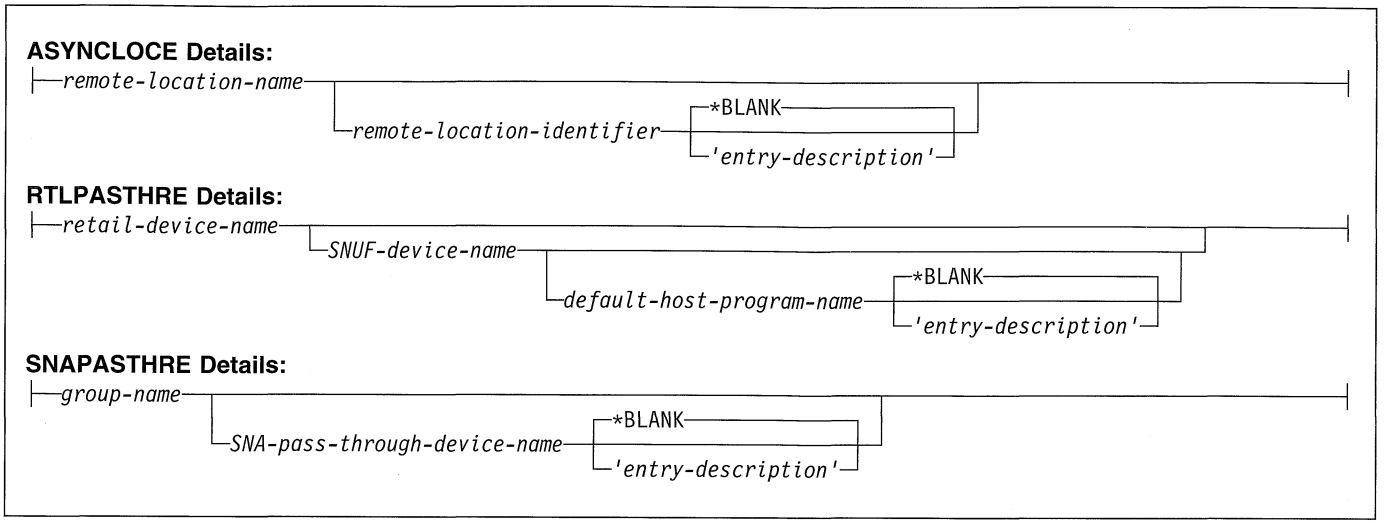
4 Location Security

5 Single Session Location

6 Locally Controlled Session

7 Preestablished Session

P All parameters preceding this point can be specified in positional form.



Purpose

The Add Configuration List Entries (ADDCFGLE) command adds entries to a configuration list.

Note: The user can also use the full screen entry display of the Change Configuration List (CHGCFGL) command to add, remove, or change entries in an existing list except for the configuration list TYPE(*SNAPASTHR).

Required Parameters

TYPE

Specifies the type of configuration list entry being added.

***APPNLCL:** An Advanced Peer-to-Peer Networking* (APPN*) local location list is used. Up to 476 APPN local location entries are allowed in the configuration list (using the CHGCFGL and CRTCFGL commands).

***APPNRMT:** An APPN remote location list is used. Up to 1898 APPN remote location entries are allowed in the configuration list (using the CHGCFGL and CRTCFGL commands).

***ASYNCADR:** An asynchronous network address list is used. Up to 294 asynchronous network address entries are allowed in the configuration list.

***ASYNCLOC:** An asynchronous remote location list is used. Up to 4995 asynchronous remote location entries are allowed in the configuration list.

***RTLPASTHR:** A retail pass-through list is used. Up to 450 retail pass-through entries can be specified in the configuration list.

***SNAPASTHR:** An SNA pass-through list is used. Up to 254 SNA pass-through entries can be specified in the configuration list.

CFGL

Specifies the name of the configuration list. This parameter is valid only when *ASYNCADR is specified on the TYPE parameter. Only one of the other configuration list types is allowed on a system. The list types have system-supplied names: QAPPNLCL, QAPPNRMT, QASYNCADR, QASYNCLOC, QRTLPASTHR, QSNAPASTHR.

Optional Parameters

APPNLCLE

Specifies the APPN local location entry. This value is required if *APPNLCL is specified for the TYPE parameter.

Up to 476 entries can be specified for this parameter, but only 50 entries can be specified at a time. An entry consists of a value from the local location name and the entry description.

Element 1: Local Location Name

local-location-name: Specify the location name of the local system. This name is used by APPN to determine whether the request being received in is for this system or another system in the network. The local location name must be unique; it cannot already exist as a remote location name that is used by configuration list QAPPNRMT and it cannot be specified on another system as a local location in the same APPN network.

Element 2: Text Description

***BLANK:** Text is not specified.

'entry-description': Specify a short description of 20 characters or less for each local entry.

APPNRMTE

Specifies the APPN remote location entry. This value is required if *APPNRMT is specified for the TYPE parameter. Up to 1898 entries can be specified for this parameter, but only 50 entries can be specified at a time. An entry consists of a value from each of the following elements. A value must be specified for each of the 11 elements for each entry..

Element 1: Remote Location Name

***ANY:** The system potentially accepts all requests sent to it.

generic-remote-location-name:* Specify the generic name (part of a name followed by an asterisk) of the remote location(s) to be changed. The generic location name allows one directory entry to be defined for all locations, on a single control point, with a name that matches the characters preceding an *.

remote-location-name: Specify the full name of a remote location.

Element 2: Remote Network Identifier

***NETATR:** The LCLNETID value specified in the system network attributes is used.

***NONE:** The remote network identifier is not specified.

remote-network-identifier: Specify the network identifier of the network in which the remote location resides.

Element 3: Local Location Name

***NETATR:** The LCLLOCNAME value specified in the system network attributes is used.

local-location-name: Specify the location name on the local system. This name is used by APPN to match a local/remote location pair entry.

Element 4: Control Point Name

***NONE:** There is no control point name.

control-point-name: Specify the name of the control point that provides network functions for the remote location. This field is required if the entry is either a generic name or *ANY.

Element 5: Control Point Network Identifier

***NETATR:** The LCLNETID value specified in the system network attributes is used.

control-point-network-identifier: Specify the network identifier in which the control point resides.

Element 6: Location Password

***NONE:** There is no location password.

location-password: Specify the password that is used when establishing sessions on the local location/remote location name pair. It must be an even number of hexadecimal characters.

Element 7: Location Security

This value specifies whether the local location allows the remote location to verify user passwords when sending program start requests to the local location.

***NO:** The remote location does not verify user passwords when sending program start requests to the local location.

***YES:** The remote location can verify user passwords when sending program start requests to the local location.

Element 8: Single Session Location

This value specifies whether the connection between the local location and remote location is a single session connection.

***NO:** A single session connection is not made between the local and the remote location.

***YES:** A single session connection is made between the local location and the remote location.

Element 9: Locally Controlled Session

This value specifies whether the single session connection between the local location and remote location is locally controlled.

***NO:** The single session connection does not have to be locally controlled.

***YES:** The single session connection is locally controlled.

Element 10: Pre-established Session

This value specifies whether the single session is automatically bound when the mode is started between the local location and remote location.

***NO:** The single session connection is not automatically made between the local and remote location.

***YES:** The single session connection is automatically made between the local and remote location.

Element 11: Remote Entry Description

***BLANK:** Text is not specified.

'entry-description': Specify a short description of 20 characters or less, enclosed in apostrophes, for each remote entry.

Element 12: Number of Single-Session Conversations

10: The number of single session conversations allowed for this device description is ten.

single-session-conversations: Specify the number of conversations allowed for a single session of this device description. Valid values range from 1 through 512.

Note: The combination of remote location name, remote network identifier, and local location name must be unique. Also, the remote location name cannot already exist as a local location in configuration list QAPPNLCL, or as the current value for either the LCLLOCNAME or the CPNAME network attribute.

ASYNCADRE

Specifies the asynchronous network address entry. This value is required if *ASYNCADR is specified for the TYPE parameter. Up to 50 entries can be specified for this parameter.

The following values make up an asynchronous network address entry.

Element 1: Network Address

network-address: Specify the asynchronous remote network address.

Element 2: Dial Retries

2: The default number of retries is two.

dial-retry: Specify the number of times dialing is retried (because errors occur) before the next number on the list is dialed. Valid values range from 1 through 255.

Element 3: Text Description

***BLANK:** Text is not specified.

'entry-description': Specify a short description of 20 characters or less for each network address entry.

ASYNCLOCE

Specifies the asynchronous remote location entry. This value is required if *ASYNCLOC is specified for the TYPE parameter. Up to 50 entries can be specified for this parameter.

The following values make up an asynchronous network address entry.

Element 1: Remote Location Name

remote-location-name: Specify the name of the remote location. This name, when combined with the remote location identifier, determines whether an incoming call is accepted. The specified name must be unique.

Element 2: Remote Location Identifier

remote-location-identifier: Specify the identifier of the remote location. When this identifier is combined with the remote location name, it determines whether an incoming call is accepted. This identifier must be the same as the remote system has for its local identifier.

Element 3: Description of Remote Location Entry

***BLANK:** Text is not specified.

'entry-description': Specify a short description of 20 characters or less for each remote location entry.

RTLPASTHRE

Specifies the retail pass-through entry. This value is required if TYPE(*RTLPASTHR) is specified. Up to 50 entries can be specified for this parameter.

The following values make up a retail pass-through entry:

Element 1: Retail Device Name

retail-device-name: Specify the name of the retail device to use for the pass-through session. This must be a unique value.

Element 2: SNA Upline Facility Device Name

SNUF-device-name: Specify the name of the host device to use for the pass-through session. This must be a unique value.

Element 3: Default Host Program Name

default-host-program-name: Specify the name of the program to be started on the host if a program name was not specified by the retail controller.

Element 4: Text Description

***BLANK:** Text is not specified.

'entry-description': Specify a short description of 20 characters or less for each retail pass-through entry.

***BLANK:** Text is not specified.

***SNAPASTHRE**

Specifies the SNA pass-through entry. This parameter is required if TYPE(*SNAPASTHR) is specified. Up to 254 entries can be specified for this parameter.

The following values make up a SNA pass-through entry:

Element 1: SNA Upstream Group Name

group-name: Specify the name of the group. This must be a unique value. The group name has host SNA pass-through device names associated with it.

Element 2: SNA Upstream Device Name

SNA-pass-through-device-name: Specify the name of the host device to use for the pass-through session.

Element 3: Text Description

***BLANK:** No description is specified.

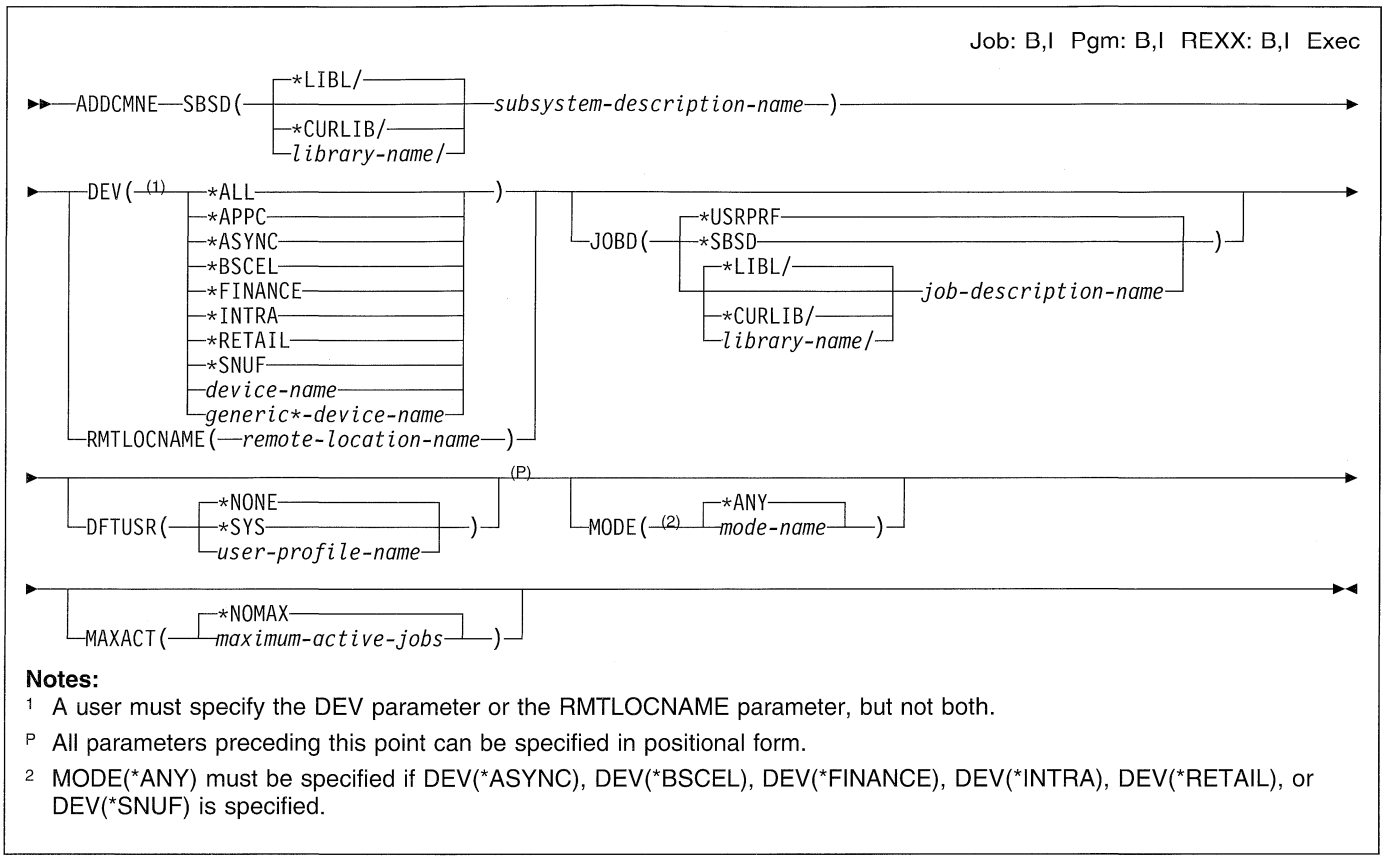
'entry-description': Specify a description of up to 50 characters for each SNA pass-through entry.

Example

```
ADDCFGLE  TYPE(*APPNLCL)  APPNLCLE((LOC1 'location one')
  (LOC2 'location two'))
```

This command adds local locations LOC1 and LOC2 to configuration list QAPPNLCL.

ADDCMNE (Add Communications Entry) Command



Notes:

1 A user must specify the DEV parameter or the RMTLOCNAME parameter, but not both.

P All parameters preceding this point can be specified in positional form.

2 MODE(*ANY) must be specified if DEV(*ASYNC), DEV(*BSCEL), DEV(*FINANCE), DEV(*INTRA), DEV(*RETAIL), or DEV(*SNUF) is specified.

Purpose

The Add Communications Entry (ADDCMNE) command adds a communications entry to an existing subsystem description. The subsystem must not be active when this command is entered.

Each entry specifies a communications device, device type, or remote location that can enter work into the subsystem.

Restriction: The user must have object operational and object management authorities for the communications entry.

Required Parameters

SBSD

Specifies the qualified name of the subsystem description to which the communications entry is being added or in which it is being changed.

The name of the subsystem description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

subsystem-description-name: Specify the name of the subsystem description to which the communications entry is added.

DEV

Specifies the name of the device description or the type of the device being used with this communications entry.

Note: A user must specify either this parameter or the RMTLOCNAME parameter, but not both.

***ALL:** All communications devices can be used with this communications entry.

***APPC:** All advanced program-to-program communications devices can be used with this communications entry. The devices created with the CRTDEVAPPC command are used by the AS/400 system APPN support.

***ASYNC:** All asynchronous communications devices, including those created with the CRTDEVASC command, can be used with this communications entry. This value is valid only if MODE(*ANY) is specified.

***BSCEL:** All binary synchronous equivalency link communications devices, including those created with the CRTDEVBSC command, can be used with this commu-

nications entry. This value is valid only if MODE(*ANY) is specified.

***FINANCE:** All FINANCE communications devices, including those created with the CRTDEVFNC command, are used with this communications entry. This value is valid only if MODE(*ANY) is specified.

***INTRA:** All INTRA communications devices, including those created with the CRTDEVINTR command, are used with this communications entry. This value is valid only if MODE(*ANY) is specified.

***RETAIL:** All RETAIL communications devices, including those created with the CRTDEVRTL command, are used with this communications entry. This value is valid only if MODE(*ANY) is specified.

***SNUF:** All SNA upline facility communications devices, including those created with the CRTDEVSNUF command, can be used with this communications entry. This value is valid only if MODE(*ANY) is specified.

device-name: Specify the device description name or the type of device to use with this communications device entry. The name specified on the CRTDEVxxx command associated with this device description name is used.

generic-device-name:* Specify the generic name of the device. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be added only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

RMTLOCNAME

Specifies the name of the remote location that is used with this object.

Note: The remote location name specified in the CRTDEVXXX command can be used here. No validity checking is done on the remote location name.

The user must specify either this parameter or the dev parameter, but not both.

Optional Parameters

JOBD

Specifies the name of the job description used. If the job description does not exist when the entry is changed, a library qualifier must be specified because the qualified job description name is retained in the subsystem description. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***USRPRF:** The job description name specified in the user profile of the user that made the program start request is used for jobs that are processed through this communications entry.

***SBSD:** The job description having the same name as the subsystem description (specified by the SBSD parameter) is used for jobs processed through this communications entry.

The name of the job description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

job-description-name: Specify the qualified name of the job description that is used for the jobs processed through this communications entry.

DFTUSR

Specifies the default user profile used for a program start request that contains no password (either all blanks or a zero length password) and user profile name. This user profile is not used for program start requests that contain a password or specify a user profile (either valid or not valid).

***NONE:** No user profile is specified as a default.

***SYS:** User program start requests are treated the same as DFTUSR(*NONE). For system-generated program start requests, the correct user profile is used.

user-profile-name: Specify the name of the user profile that is used for all program start requests that enter the system through this communications entry and that contain no password or user profile name.

Note: The names QSECOFR, QSPL, QDOC, QDBSHR, QRJE, and QSYS are not valid entries on this parameter.

MODE

Specifies the mode name of the communications device or remote location name used with this communications entry.

***ANY:** Any available modes defined to the communications device or remote location are allocated to the subsystem. If the communications device does not have defined modes, the communications device itself is allocated to the subsystem.

mode-name: Specify a mode name of the communications device or remote location used with this communications entry.

MAXACT

Specifies the maximum number of program start requests that can be active at the same time through this communications entry. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***NOMAX:** There is no maximum number of jobs that can be active at the same time.

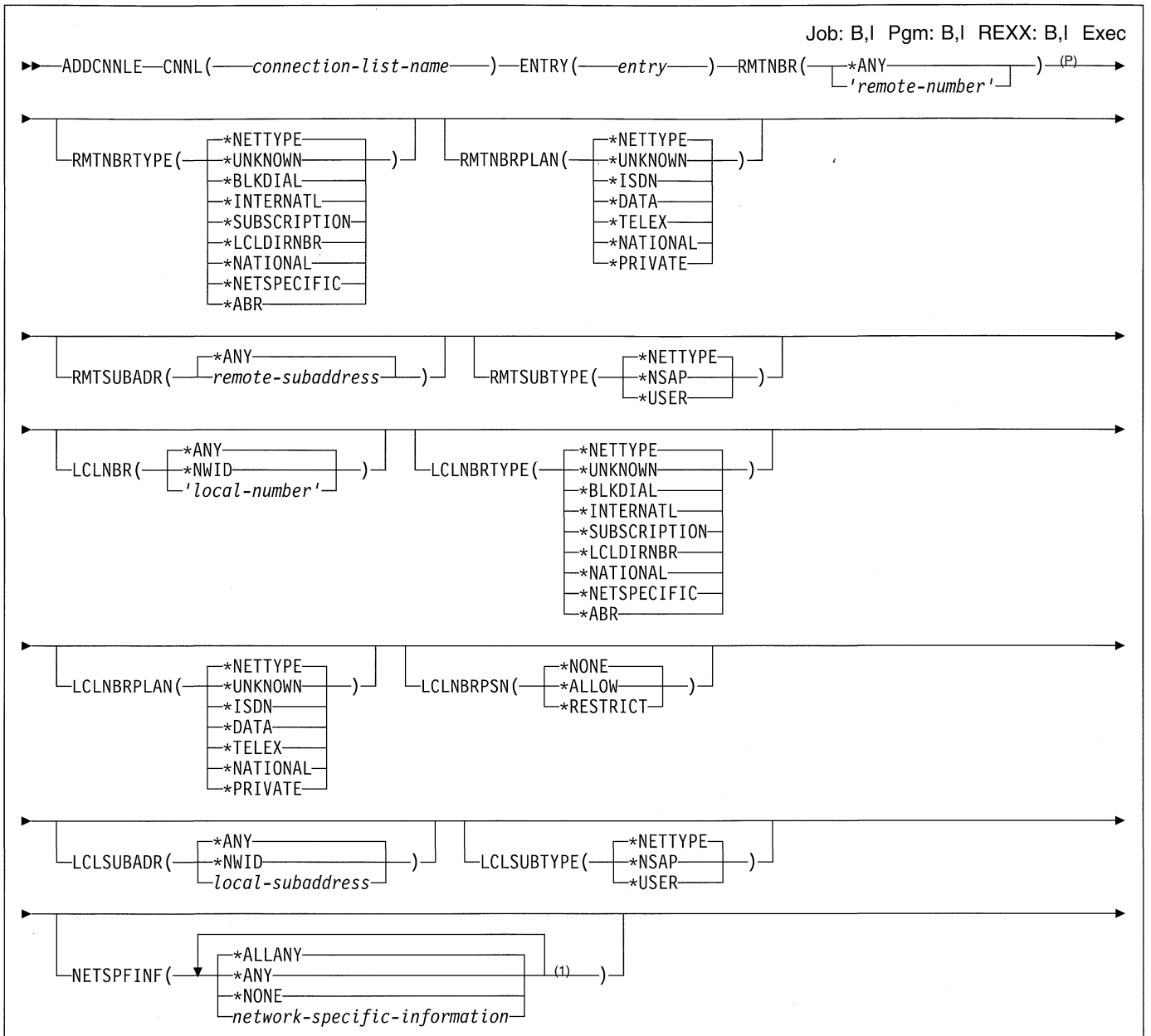
maximum-active-jobs: Specify the maximum number of jobs that can be active at the same time through this communications entry.

Example

```
ADDCMNE    SBSD(ALIB/SBS1)  DEV(COMDEV)
```

This command adds a communications entry for the APPC device named COMDEV to the subsystem description SBS1, which resides in library ALIB. The DFTUSR parameter defaults to *NONE, which means that no jobs may enter the system through this entry unless valid security information is supplied on the program start request.

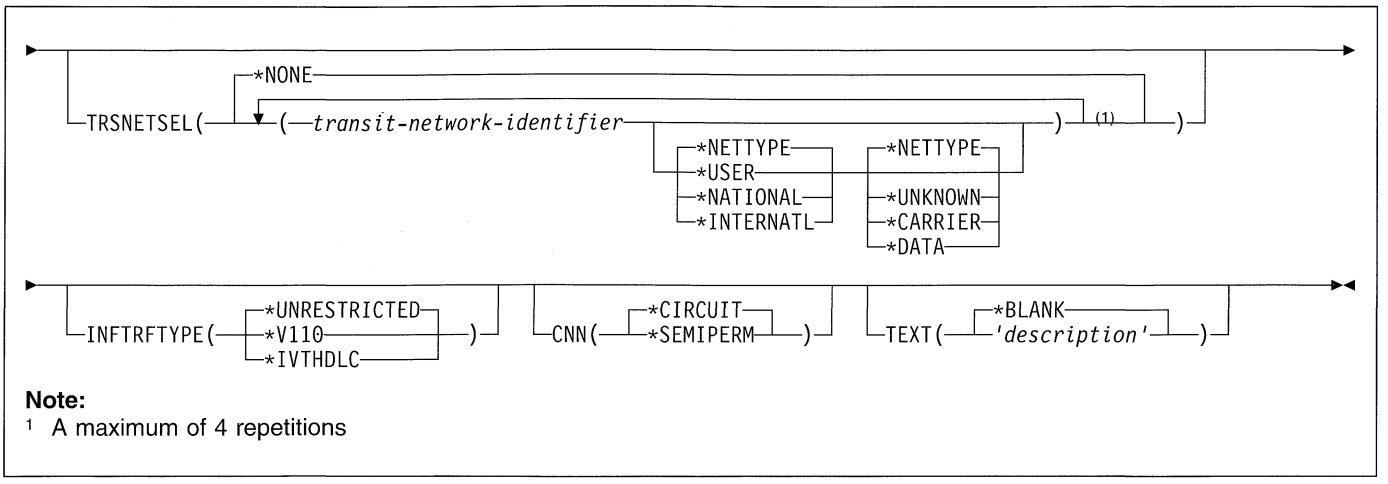
ADDCNNLE (Add Connection List Entry) Command



Notes:

1 A maximum of 4 repetitions

P All parameters preceding this point can be specified in positional form.



Note:
1 A maximum of 4 repetitions

Purpose

The Add Connection List Entry (ADDCNNLE) command is used to add an entry to a connection list.

Required Parameters

CNNL
Specifies the name of the connection list to which this entry is added.

ENTRY
Specifies the name of the entry to be added to the connection list. Each entry in the connection list must have a unique name.

RMTNBR
Specifies the number of the remote system in the Integrated Services Digital Network (ISDN).

***ANY:** Any value, including no value, specified in the received Called Party Number information element (IE) (encoded on the call by the system) is acceptable for incoming calls. For outgoing calls, the system requires the target number to be supplied so that a Called Party Number or Keypad Facility IE can be encoded on the outgoing call. Therefore, if *ANY is specified for an outgoing call, the call out attempt fails.

'remote-number': Specify the remote number up to 40 characters enclosed in apostrophes. Extra characters, such as parentheses, can be used to aid in readability, if they are specified on the remove character (RMVCHR) parameter when the connection list was created using the Create Connection List (CRTCNNL) command.

Optional Parameters

RMTNBRTYPE
Specifies the type of remote number.

***NETTYPE:** A default value is used based on the network type specified on the NETTYPE parameter when the connection list was created using the CRTCNNL command.

***UNKNOWN:** The remote number type is unknown.

***BLKDIAL:** The remote number type is unknown. *BLKDIAL is used in France instead of *UNKNOWN.

***INTERNATL:** The remote number is an international number type.

***SUBSCRIPTION:** The remote number is a subscription number type.

***LCLDIRNBR:** The remote number is a subscription number type. *LCLDIRNBR is used by AT&T** versions 5E5 and 5E6 instead of *SUBSCRIPTION.

***NATIONAL:** The remote number is a national address type.

***NETSPECIFIC:** The remote number type is specific to the network.

***ABR:** The remote number type is abbreviated.

RMTNBRPLAN
Specifies the numbering plan used for the remote number.

***NETTYPE:** A default value is used for the parameter based on the network type specified on the NETTYPE parameter when the connection list was created using the CRTCNNL command.

***UNKNOWN:** The numbering plan is not known.

***ISDN:** The ISDN/telephony numbering plan is used.

***DATA:** The data numbering plan is used.

***TELEX:** The Telex** numbering plan is used.

***NATIONAL:** The national numbering plan is used.

***PRIVATE:** A private numbering plan is used.

RMTSUBADR
Specifies the subaddress of the remote system.

***ANY:** Any value, including none, specified in the received Called Party Subaddress IE is acceptable for incoming calls. For outgoing calls, no Called Party Subaddress IE is encoded.

remote-subaddress: Specify the subaddress of the remote system; up to 40 hex characters.

RMTSUBTYPE
Specifies the type of remote subaddress.

***NETTYPE:** A default value is used based on the network type specified on the NETTYPE parameter when the connection list was created using the CRTCNNL command.

***NSAP:** The remote subaddress type is NSAP (X.213).

***USER:** Some of the common OS/400 objects and database physical files are included.

LCLNBR
Specifies information about the local number that is called for an incoming call. If the entry is used for an outgoing call this parameter is ignored.

***ANY:** Any value, including no value, specified in the received Called Party Number IE is acceptable.

***NWID:** Any value, including no value, specified in the received Called Party Number IE is acceptable. The number used for outgoing calls is determined by the network interface description.

'local-number': Specify the local number (up to 40 hex characters enclosed in apostrophes). For this entry, only calls directed at this local number are accepted. Extra characters, such as parentheses, can be used if they are specified on the remove character parameter (RMVCHR parameter).

LCLNBRTYPE
Specifies the type of local number.

***NETTYPE:** The type is determined by the value specified on the *NETTYPE parameter when the connection list was created using the CRTCNNL command.

***UNKNOWN:** The local number type is unknown.

***BLKDIAL:** The remote number type is unknown. *BLKDIAL is used in France instead of *UNKNOWN.

***INTERNATL:** The local number is an international number type.

***SUBSCRIPTION:** The local number is a subscription number type.

***LCLDIRNBR:** The local number is a subscription number type. *LCLDIRNBR is used by AT&T versions 5E5 and 5E6 instead of *SUBSCRIPTION.

***NATIONAL:** The local number is a national address type.

***NETSPECIFIC:** The local number type is specific to the network.

***ABR:** The local number type is abbreviated.

LCLNBRPLAN
Specifies the numbering plan used for the local number.

***NETTYPE:** A default value is used based on the network type specified on the NETTYPE parameter when the connection list was created using the CRTCNNL command.

***UNKNOWN:** The numbering plan is not known.

***ISDN:** The ISDN/telephony numbering plan is used.

***DATA:** The data numbering plan is used.

***TELEX:** The telex numbering plan is used.

***NATIONAL:** The national numbering plan is used.

***PRIVATE:** A private numbering plan is used.

LCLNBRPSN
Specifies the intention of the calling user for the presentation of the local number to the called user. This parameter applies only to outgoing calls.

***NONE:** The local number presentation is not encoded. The network determines whether to present the local number to the called user.

***ALLOW:** The local number is presented to the called user.

***RESTRICT:** The presentation of the local number to the called user is restricted by the network.

LCLSUBADR
Specifies the subaddress of the local system.

***ANY:** Any value, including no value, specified in the Received Calling Party Subaddress IE is acceptable for incoming calls. For outgoing calls, no Called Party Subaddress IE is encoded.

***NWID:** Any value, including no value, specified in the received Called Party Subaddress IE is acceptable for incoming calls. For outgoing calls, the Called Party Subaddress IE is encoded on the network interface description.

local-subaddress: Specify the local subaddress; up to 40 hex characters.

LCLSUBTYPE
Specifies the type of subaddress used by the local system.

***NETTYPE:** A default value is used based on the network type specified on the NETTYPE parameter when the connection list was created using the CRTCNNL command.

***NSAP:** The subaddress type is NSAP (X.213).

***USER:** The subaddress type is user-specified.

NETSPFINF
Specifies network-specific information.

Network-specific information requests special network facilities on an outgoing call or for the network to inform the system of special facilities used on incoming calls. These special facilities and the encoding of the network-specific information are network-specific. If the user wants to use the network-specific information, contact the network provider for information on how to encode it. Some networks can process outgoing calls with this

information, but fail to deliver the calls to the remote user. Contact the user's network provider to determine if this is the case.

The user can enter up to four network-specific information values. Values are processed in chronological order: values of *NONE are automatically moved to the last network-specific information values. The order of the network-specific information values in the outgoing call at the local system must match the order expected at the remote system for incoming calls.

***ALLANY:** The *ANY value is used for all network-specific information elements. *ALLANY can be specified only once for this parameter

***ANY:** Any value, including *NONE, specified in the received network-specific facility IE is acceptable for incoming calls. If the entry is used for an outgoing call, no network-specific information is sent.

Note: *ANY is the default for each network-specific information value,

***NONE:** It is unacceptable to have network-specific information on an incoming call for this entry. If the entry is used for an outgoing call, no network-specific information is sent.

network-specific-information: Specify, in even numbers, the network-specific information up to 60 hex characters. It is encoded into the network-specific facility IE as follows:

```
x'40' Length<network-specific information)>)
<network-specific information>
```

TRSNETSEL

Specifies up to four Transit Network Selections. The selection information consists of three parts: 1) a transit network identifier, 2) a transit network type, and 3) a transit network plan. The information for the transit network selection is used only when outgoing calls are placed.

The purpose of the Transit Network Selection is to identify the intermediate ISDNs that must be followed by the ISDN to reach the remote user. The encoding of this information is network-specific. Contact the network provider for information on how the Transit Network Selection information should be encoded.

***NONE:** No Transit Network Selection information is sent on the outgoing call. *NONE is the default value for each Transit Network Selection values.

Element 1: Transit Network Identifier

transit-network-identifier: Specify the transit network identifier.

Element 2: Transit Network Type

***NETTYPE:** A default value is used based on the network type specified on the NETTYPE parameter when the connection list was created using the CRTCNNL command.

***USER:** The transit network type is user-specified.

***NATIONAL:** The transit network is a national type.

***INTERNATL:** The transit network is an international type.

Element 3: Transit Network Plan

***NETTYPE:** A default value is used based on the network type specified on the NETTYPE parameter when the connection list was created using the CRTCNNL command.

***UNKNOWN:** The network plan is unknown.

***CARRIER:** The carrier identification code plan is used.

***DATA:** The data identification code (X.121) plan is used.

INFTRFTYPE

Specifies the information transfer type. The information transfer type determines the layer-1 protocol.

***UNRESTRICTED:** The data-channel traffic appears as digital information; no physical transformation is required and each B-channel operates at capacity (64k bits per second (bps)).

***V110:** The transfer type is V-series Recommendation 110. Each B-channel operates at 56k bps.

***IVTHDLC:** The transfer type is Inverted HDLC. Each B-channel operates at capacity, 64k bps.

CNN

Specifies the type of line connection used.

***CIRCUIT:** The entry is for a circuit-switched connection.

***SEMIPERM:** The entry is for a semi permanent connection.

TEXT

Specifies text that briefly describes the entry in the connection list. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***BLANK:** Text is not specified.

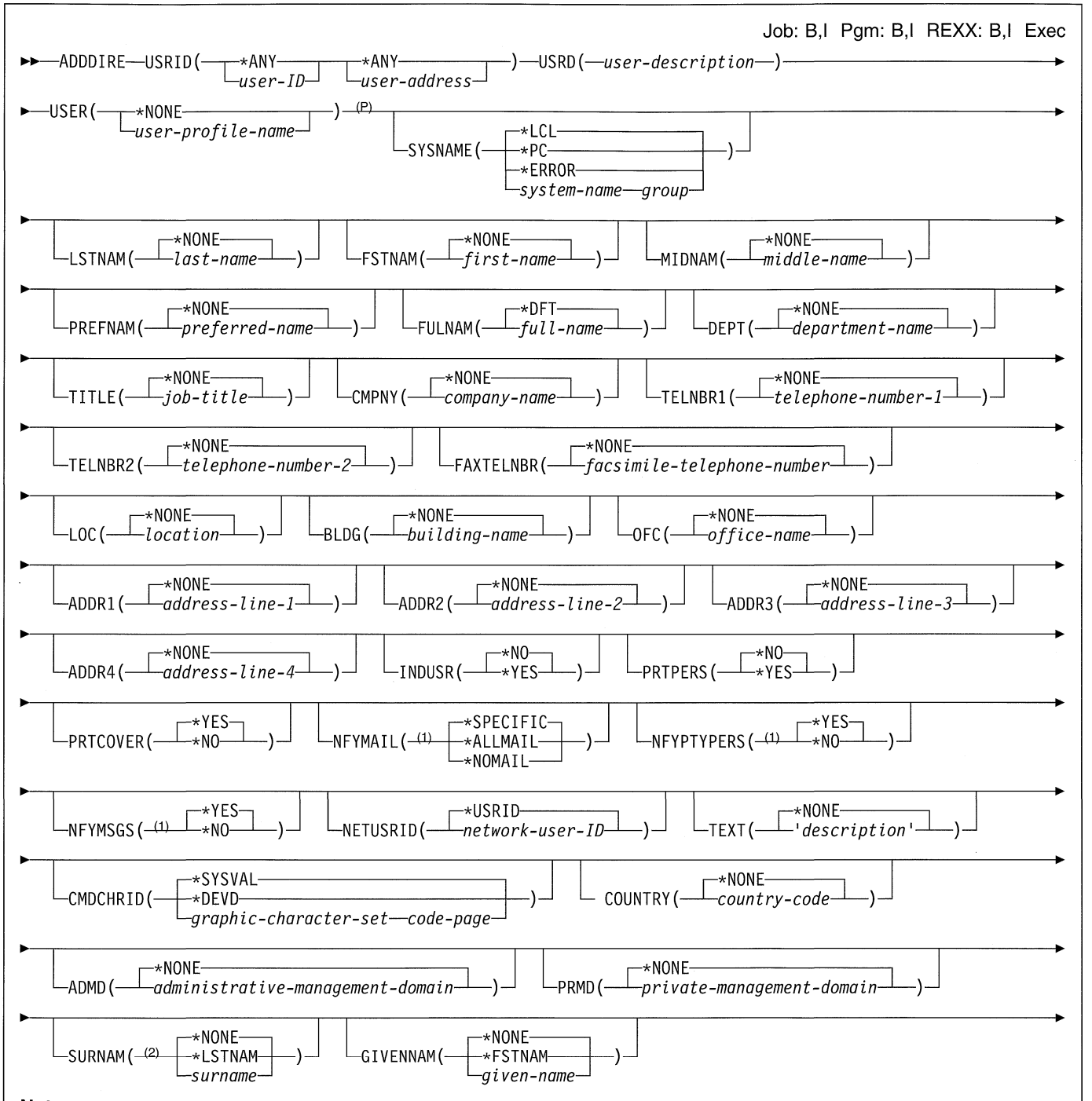
'description': Specify a description of the entry. Specify up to 50 characters of text, enclosed in apostrophes.

Example

```
ADDCNNLE  CNNL(CHICAGO)  ENTRY(CORPORATE)
  RMTNBR('(896) 999-5555')
```

This command adds entry CORPORATE in connection list CHICAGO. The entry will have (896) 999-5555 as a remote number.

ADDDIRE (Add Directory Entry) Command

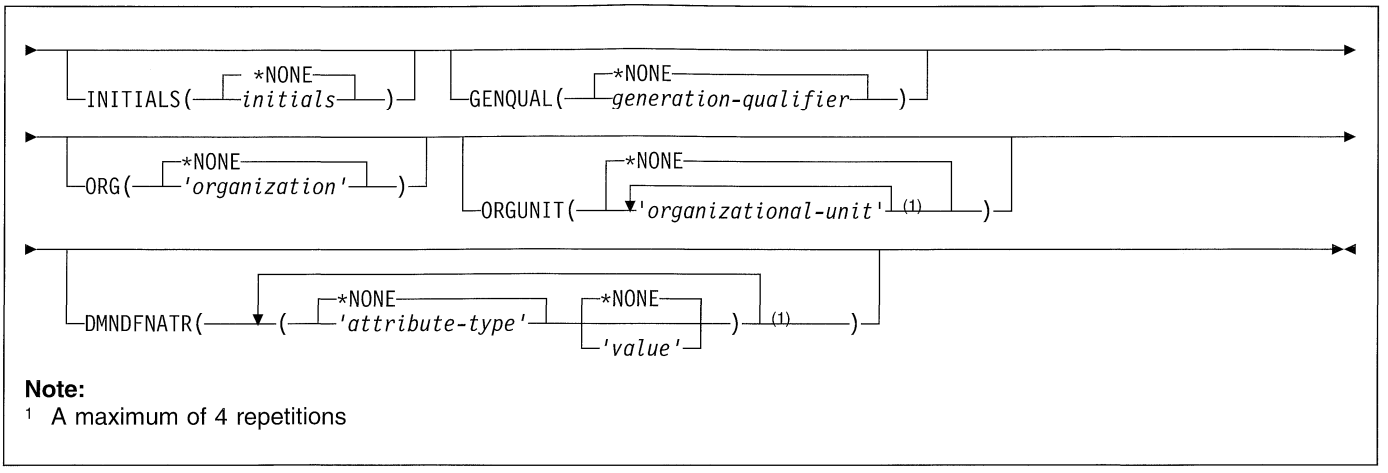


Notes:

P All parameters preceding this point can be specified in positional form.

1 NFYPTYPERS and NFYMSGS do not apply unless NFYMAIL(*SPECIFIC) is specified.

2 SURNAM is required when GIVENAME, INITIALS, or GENQUAL is specified.



Note:
1 A maximum of 4 repetitions

Purpose

The Add Directory Entry (ADDDIRE) command is used to add new entries to the system distribution directory. The directory contains the user's user identifier (ID), profile name, system name, system address, mailing address, telephone numbers, and other user information. The ADDDIRE command provides a parameter for each of the fields contained in the directory.

The ADDDIRE command does not provide interactive display support. This is provided by the Work with Directory (WRKDIR) command.

An X.400 originator/recipient (O/R) name can be added to the directory with this command. X.400 is an international standard for communications and the O/R name is the addressing information used in X.400 communications. The X.400 O/R name must be in character set 1169 and code page 500. This set includes A through Z, 0 through 9, and some special characters. Additional information on characters allowed is in the *National Language Support Planning Guide*.

Notes:

1. To prevent the system from changing lowercase characters to uppercase, enclose the values in apostrophes. This does not apply to user ID/address, system name/group, department, or X.400 O/R name.
2. Only the user ID/address, system name/group, department, and X.400 O/R name are translated from the graphic character identifier (GCID) specified by the CMDCHRID parameter. All other parameters are stored exactly as they are entered and the GCID is stored with them. The default GCID value is taken from the QCHRID system value. The user can override the defaults by specifying a character set and code page or by specifying *DEVD for the display device description.
3. Double-byte character set (DBCS) characters can be entered for the following system directory entry parameters:

USRD	LOCATION
LSTNAM	BLDG
FSTNAM	OFC
MIDNAM	ADDR1
PREFNAM	ADDR2
FULNAM	ADDR3
DEPT	ADDR4
TITLE	TEXT
CMPNY	

Restriction: The user of this command must have security administrator authority.

Required Parameters

USRID

Specifies the user ID and address of the user for whom the directory entry is made. Both elements must be specified. If lowercase characters are specified, the system translates and stores them as uppercase characters. Further information about specifying the user ID and address is in the *Distribution Services Network Guide*.

Element 1: User ID

***ANY:** Any user ID at the address specified on Element 2 of this parameter is used. Only one *ANY is allowed for each address. This value is used to resolve a distribution that does not match a specific user ID but matches an address.

user-ID: Specify the user ID for this directory entry. Up to 8 characters can be specified. If this value is specified, an address must be specified for Element 2.

Element 2: User Address

***ANY:** Any address for the user ID specified on Element 1 is used. One **USRID(*ANY *ANY)** entry is allowed in the directory. This value is used to resolve distributions that do not match any other directory entries.

user-address: Specify the address for this directory entry. Up to 8 characters can be specified.

USRD

Specifies the description associated with the user ID and address. For example, the description can contain the user's full name, department number, or position. Specify up to 50 characters for the description.

USER

Specifies the user profile of the user being added to the directory. If the user being added is a local user, a valid profile must exist on the local system.

A user profile is required to define a local user whose mail is sent to the remote system specified in the system name/group.

***NONE:** The user being added to the directory is a remote user and has no local profile. If *NONE is specified and the SYSNAME parameter indicates the local system, an error message is returned.

user-profile-name: Specify a valid system user profile name up to 10 characters in length. The user profile name is required for all local users. If a profile name is specified for a user whose mail is sent to a remote system, the profile name must be valid on the local system.

Optional Parameters

SYSNAME

Specifies the one- or two-part name of the system on which the user works. If a two-part system name is specified on the command line, up to 8 characters make up both the system name and the system group name. The parts should be separated by at least one space.

A remote user can be added to the directory before the system name and system group are defined in the network tables, but distributions cannot be sent to that user until the remote system name and system group are defined. The remote system name and system group name are defined by using the Configure Distribution Services (CFGDSTSRV) command. Additional information on defining a remote system name and group name is in the *Distribution Services Network Guide*.

***LCL:** The system name is its local name. All local users being added to the directory should have *LCL specified as the system name.

***PC:** This system name is for distributed systems node executive (DSNX) users with a personal computer (PC) attached to the system.

***ERROR:** This value is used when the user's network contains a central system that receives all unresolved distributions. In this type of network, distribution looping may be encountered when a distribution cannot find a specific user ID on the intended system and the intended system has an *ANY *ANY entry directing distributions to the central system. The central system also has a default *ANY *address* entry directing unresolved distributions to the intended system. To prevent distribution looping, specify *ERROR as the system name for the default entry being added. When a distribution cannot find a specific user ID, but matches this default entry, the distribution is handled as a user that is not valid, just as if no directory match were found.

*ERROR is valid only when USRID(*ANY *address*) or USRID(*ANY *ANY) is specified.

Element 1: System Name

system-name: Specify the name of the system on which the user works.

Element 2: System Group Name

group: Specify the system group name of the system on which the user works.

LSTNAM

Specifies the user's last name. If no names are provided (last, first, middle, preferred, or full), but a value is specified on the DEPT parameter, the last name will default to an asterisk (*). This is because the directory department function requires a non-blank full name when a department value is specified.

***NONE:** No last name is specified.

last-name: Specify up to 40 characters for the user's last name.

FSTNAM

Specifies the user's first name.

***NONE:** No first name is specified.

first-name: Specify up to 20 characters for the user's first name.

MIDNAM

Specifies the user's middle name.

***NONE:** No middle name is specified.

middle-name: Specify up to 20 characters for the user's middle name.

PREFNAM

Specifies the user's preferred name. For example, "Jonathan" likes to be called "Jon."

***NONE:** No preferred name is specified.

preferred-name: Specify up to 20 characters for the user's preferred name.

FULNAM

Specifies the user's full name. Directory entries are shown in the full name format when using the search and department functions. It is recommended that the user institute a consistent naming convention for the full name. Note that uppercase and lowercase alphabetic characters have different sorting sequences. Making the first character of each name uppercase and the rest that follow lowercase is the preferred format.

If FULNAM(*DFT) is specified, the following format is used to create the full name:

```
LAST NAME, FIRST NAME  MIDDLE NAME (PREFERRED NAME)
```

The preferred name is always enclosed in parentheses. If no user-defined values are specified for the last, first, and middle names, but the DEPT parameter contains a value, the last name defaults to an asterisk (*). If the user specifies FULNAM(*DFT), the full name defaults to an asterisk because it is built from the last name.

***DFT:** The full name is created from the user-defined values specified on the LSTNAM, FSTNAM, MIDNAM, and PREFNAM parameters.

full-name: Specify up to 50 characters for the user's full name.

DEPT

Specifies the name of the department of which the user is a member.

***NONE:** The user is not defined as a member of a department.

department-name: Specify up to 10 characters for the name of the user's department.

TITLE

Specifies the user's job title.

***NONE:** No job title is added.

job-title: Specify up to 40 characters for the user's job title.

CMPNY

Specifies the name of the user's company.

***NONE:** No company name is added.

company-name: Specify up to 50 characters for the name of the user's company.

TELNBR1

Specifies the primary telephone number of the user. The telephone number can be specified in any arrangement appropriate for the user, including an international telephone number.

***NONE:** No primary telephone number is specified.

telephone-number-1: Specify up to 26 characters for the primary telephone number of the user.

TELNBR2

Specifies a second telephone number for the user. The telephone number can be specified in any arrangement appropriate for the user, including an international telephone number.

***NONE:** No second telephone number is specified.

telephone-number-2: Specify up to 26 characters for the second telephone number of the user.

FAXTELNBR

Specifies a facsimile telephone number for the user. The facsimile telephone number can be specified in any arrangement appropriate for the user, including an international telephone number.

***NONE:** No facsimile telephone number is specified.

facsimile-telephone-number: Specify up to 32 characters for the user facsimile telephone number.

LOC

Specifies the location of the user. For example, the location can specify a building and floor, a department, or a remote site.

***NONE:** No location is specified.

location: Specify up to 40 characters for the location of the user.

BLDG

Specifies the name of the building in which the user works.

***NONE:** No building name is added.

building-name: Specify up to 20 characters for the name of the building in which the user works.

OFC

Specifies the name of the office in which the user works.

***NONE:** No office name is added.

office-name: Specify up to 16 characters for the name of the office in which the user works.

ADDR1-ADDR4

These four parameters specify the mailing address of the user. Up to 40 characters of data can be specified in each of these fields.

***NONE:** No address lines are specified.

address-line-1-4: Specify the user's mailing address in any arrangement, up to 40 characters per line.

INDUSR

Specifies whether the user being added to the directory is an indirect user. An indirect user is a local user who does not sign on the system to receive mail but receives printed mail. An indirect user is a local user and must have a profile on the local system.

***NO:** The user is not an indirect user.

***YES:** The user is an indirect user.

PRTPERS

Specifies whether personal mail for an indirect user is printed. Consideration should be given to restricting public access to the printer when personal mail is printed.

***NO:** No personal mail is printed for this indirect user.

***YES:** Personal mail is printed for this indirect user.

PRTCOVER

Specifies whether a cover page is printed when the user's mail is printed.

***YES:** The cover page is printed.

***NO:** The cover page is not printed.

NFYMAIL

Specifies whether the user is notified of the arrival of mail.

***SPECIFIC:** The user is notified of the arrival of specific types of mail. The types of mail are specified on the NFYPTYPERS parameter and the NFYMSGS parameter.

***ALLMAIL:** The user is notified of the arrival of all types of mail.

***NOMAIL:** The user is not notified of the arrival of mail.

NFYPTYPERS

Specifies whether the user is notified of the arrival of priority and personal mail.

Note: This parameter is valid only if NFYMAIL(*SPECIFIC) is specified.

***YES:** The user is notified of the arrival of priority and personal mail.

***NO:** The user is not notified of the arrival of priority and personal mail.

NFYMSGS

Specifies whether the user is notified of the arrival of messages.

***YES:** The user is notified of the arrival of messages.

***NO:** The user is not notified of the arrival of messages.

NETUSRID

Specifies the unique network user ID for this directory entry. This ID is used during directory shadowing to uniquely identify a user in a network.

***USRID:** Set the network user ID to the user ID and address associated with this directory entry.

network-user-ID: Specify the network user ID for the user. A maximum of 47 characters can be specified.

TEXT

Specifies any additional information to describe the directory entry. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***NONE:** No text is specified.

'description': Specify up to 50 characters of text to describe additional information about the user.

CMDCHRID

Specifies the character identifier (graphic character set and code page) for data being specified as parameter values on this command. This character identifier (CHRID) is related to the display device used to specify the command. More information about CHRID processing is in the *Guide to Programming Displays*.

***SYSVAL:** The system determines the graphic character set and code page values for the command parameters from the QCHRID system values.

***DEVD:** The system determines the graphic character set and code page values for the command parameters from the display device description where this command is entered. This option is supported only when the command is entered from an interactive job. If this option is specified in a batch job, an error message is returned.

Element 1: Character Set

graphic-character-set: Specify the character set used to create the command parameters. Valid values range from 1 through 9999 characters.

Element 2: Code Page

code-page: Specify the code page. Valid values range from 1 through 9999.

COUNTRY

Specifies the country code part of the X.400 O/R name.

***NONE:** No country code is specified.

country-code: Specify an ISO 3166 Alpha-2 code or a CCITT country code. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

ADMD

Specifies the administrative management domain part of the X.400 O/R name.

***NONE:** No administrative management domain is specified.

administrative-management-domain: Specify a 1- to 16-character administrative management domain name. An administrative management domain is a public organization that handles a management domain. A management domain is a set of message transfer agents and user agents that comprise a system capable of handling messages.

PRMD

Specifies the private management domain part of the X.400 O/R name.

***NONE:** No private management domain is specified.

private-management-domain: Specify a 1- to 16-character description of the private management domain. A private management domain is a private company or a non-commercial organization that handles a management domain. A management domain is a set of message transfer agents and user agents that comprise a system capable of handling messages.

SURNAM

Specifies the X.400 last name part of the personal name within the X.400 O/R name.

***NONE:** No surname is specified.

***LSTNAM:** The last name of the user specified in the directory entry is used as the surname.

surname: Specify up to 40 characters for the surname.

GIVENNAM

Specifies the X.400 user first name part of the personal name within the X.400 O/R name.

***NONE:** No given name is specified.

***FSTNAM:** The user first name specified in the directory entry is used as the given name. It is truncated to 16 characters.

given-name: Specify up to 16 characters for the given name.

INITIALS

Specifies the initials part of the personal name within the X.400 O/R name. For example, the initials for John Henry Smith are JH.

***NONE:** No initials are specified.

initials: Specify up to 5 characters for the initials.

GENQUAL

Specifies the generation qualifier part of the personal name within the X.400 O/R name.

***NONE:** No generation qualifier is specified.

generation-qualifier: Specify up to 3 characters for the generation qualifier.

ORG

Specifies the organization part of the X.400 O/R name.

***NONE:** No organization name is specified.

'organization': Specify an organization name of up to 64 characters.

ORGUNIT

Specifies the organization-defined unit part of the X.400 O/R name.

***NONE:** No organizational unit is specified.

'organizational-unit': Specify up to 32 characters for the name of an organizational unit. Up to 4 organizational units can be listed in order of descending significance.

DMNDFNATR

Specifies the type and value of a domain-defined attribute not specified by X.400 standards but allowed in the X.400 O/R name to accommodate existing message handling systems. Up to 4 sets of attributes can be specified.

Element 1: Domain-Defined Attribute Type

***NONE:** No domain-defined attribute type is specified.

'attribute-type': Specify up to 8 characters for the description of the domain-defined attribute type.

Element 2: Domain-Defined Attribute Value

***NONE:** No domain-defined attribute value is specified.

'value': Specify up to 128 characters for the description of the domain-defined attribute value.

Examples

Example 1: Adding a Local User

```
ADDDIRE   USRID(HURST PAYROLL)
  USRD('Manager of Payroll')
  USER(ABHURST)
  LSTNAM(Hurst)
  FSTNAM(Arthur)
  PREFNAM(Art)
  DEPT(55K)
  ADDR1('Dept55K/025-3')
  ADDR2('IBM Rochester')
  ADDR3('Highway 52 North')
  ADDR4('Rochester, MN 55904')
  LOC('Main Office')
  BLDG(025-3)
  OFC(E219)
  TELNBR1('435-422-2120')
  TELNBR2('435-422-1012')
  FAXTELNBR('435-422-3296')
```

This command adds a local user to the distribution directory by allowing the system name parameter to default to *LCL. Since this is a local user, the user profile is specified. Address lines, location, and telephone numbers have been specified. Since the TEXT parameter is not used, it defaults to *NONE. This user is not an indirect user since the INDUSR parameter defaulted to *NO.

The user's last, first, and preferred names are specified. The full name was not specified, so FULNAM(*DFT) is used and will be created as, 'Hurst, Arthur (Art)'. This user has been added as a member of the department named 55K. If this department is searched, then 'Hurst, Arthur (Art)' will be included on the search list.

The user entry is added to the directory if each one of the following is true:

1. A user ID and address HURST PAYROLL is not already in the directory.
2. The user profile name ABHURST is not already in the directory.

Example 2: Adding a Remote User

```
ADDDIRE   USERID(BYRD NEWYORK)
  USRD('Arthur J. Byrd')
  USER(*NONE) SYSNAME(BOCA)
  LOC('Boca Raton, Florida')
  DEPT(61Q)
```

This command adds a remote user entry to the distribution directory. Since this is a remote user, the USER(*NONE) parameter is specified. The system name without the system group is specified. Except for the location, all of the parameters use default values. If the user-ID and address are unique, the user entry is added to the directory as a remote user.

If you are using directory shadowing, you do not need to add remote users as these users can be shadowed to your system.

If a department value is specified for this user, but no user name is specified, the last name is set to '*'. The full name is also '*' because it is created from the last name. This is done because the directory requires a non-blank name with department.

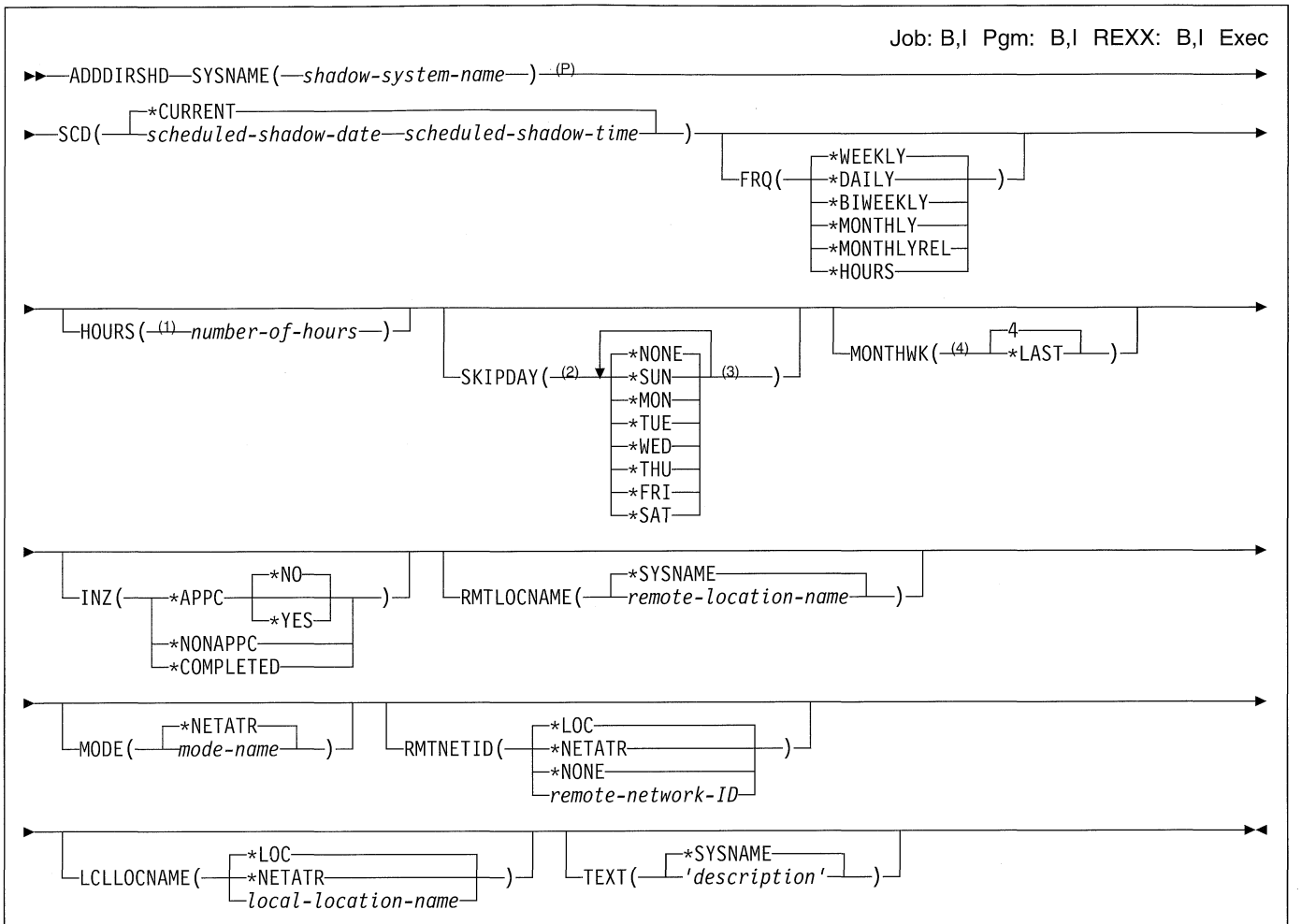
Additional Considerations

Multiple descriptions can be associated with a given user ID and address. For example, HURST DEPT48K can have an entry with the description Arthur B. Hurst and an entry with the description Manager of Dept. 48K. The ADDDIRE command does not support adding another description to a user-ID. The Work with Directory (WRKDIR) is used for adding multiple descriptions for a user.

For local users, there is a one-to-one correspondence between the user ID and address and the user profile. Only one user ID and address can be associated with a user profile name, and only one user profile name can be associated with a user ID and address. If a user profile name is specified on the ADDDIRE command that is already associated with an existing user ID and address in the directory, an error message is returned.

This should not cause a problem for remote users since the user profile name is not specified. However, if the profile is specified, it is verified to determine that the profile name is not already in the directory. If the profile is in the directory, an error message is returned.

ADDDIRSHD (Add Directory Shadow System) Command



Notes:

1 Valid only when FRQ(*HOURS) is specified.

2 Valid only when when FRQ(*DAILY) is specified.

3 A maximum of 5 repetitions.

4 Valid only when FRQ(*MONTHLYREL) is specified and when the date specified on the SCD parameter is the 22nd, 23rd or 24th.

P All parameters preceding this point can be specified in positional form.

Purpose

The Add Directory Shadow System (ADDDIRSHD) command adds a supplier system to supply system distribution directory data to your system through directory shadowing.

Restriction: To use this command, you must have security administrator (*SECADM) authority.

Required Parameters

SYSNAME

Specifies a maximum of 8 characters for the name of the supplier system you are adding. You can specify uppercase letters A through Z, numbers 0 through 9, and special characters @, #, $, and embedded blanks. Embedded blanks must be enclosed in single quotation marks ('). Leading blanks are not allowed. The @, #, and $ characters are not recommended because they are not part of an invariant character set and are not available on all keyboards.

SCD

Specifies the date and time at which the system you are adding begins supplying data to your system.

***CURRENT:** The system begins supplying data at the current date and time.

Element 1: Shadow Date

scheduled-shadow-date: Specify the date on which the system begins supplying data to your system. The date must be specified in the job date format.

Element 2: Shadow Time

scheduled-shadow-time: Specify the time at which the system begins supplying data to your system.

The time is specified in 24-hour format with or without a time separator as follows:

- With a time separator, specify a string of 5 or 8 digits where the time separator separates the hours, minutes, and seconds. If this command is entered from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command fails.
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds. Valid values for **hh** range from 00 through 23. Valid values for **mm** and **ss** range from 00 through 59.

Optional Parameters

FRQ

Specifies the frequency with which the supplier system you are adding shadows data to your system, based on the value specified on the SCD parameter.

***WEEKLY:** Shadowing occurs once a week.

***DAILY:** Shadowing occurs once a day.

***BIWEEKLY:** Shadowing occurs every other week.

***MONTHLY:** Shadowing occurs on the same date every month.

***MONTHLYREL:** Shadowing occurs on the same relative day of the same relative week of every month, such as the first Monday of the month.

***HOURS:** Shadowing occurs in the interval specified on the HOURS parameter.

HOURS

Specifies the number of hours between shadows from the supplier system you are adding.

SKIPDAY

Specifies, when FRQ(*DAILY) is specified, the days of the week when shadowing does not occur. A maximum of five values, other than *NONE, can be specified.

***NONE:** No days are skipped.

***SUN:** Sundays are skipped.

***MON:** Mondays are skipped.

***TUE:** Tuesdays are skipped.

***WED:** Wednesdays are skipped.

***THU:** Thursdays are skipped.

***FRI:** Fridays are skipped.

***SAT:** Saturdays are skipped.

MONTHWK

Specifies whether shadowing that occurs on the same relative day of the month is scheduled to occur in the fourth week or the last week of the month.

4: Shadowing occurs on the same relative day in the fourth week of the month.

***LAST:** Shadowing occurs on the same relative day in the last week of the month, whether or not the month has four or five weeks.

INZ

Specifies the method used for the first shadow from the supplier system. The first shadow duplicates all of the local and shadowed data in the supplier system's distribution directory. Remote users are optionally supplied when the supplier specifies RMTSHD(*YES) on the Change Directory Attributes (CHGDIRA) command. Subsequent shadows include only data that has changed since the previous shadow.

Element 1: Automatic Shadow

***APPC:** The first shadow occurs when this command is run using advanced program-to-program (APPC) communications. If you are adding a supplier system with a large directory, you may want to specify *NONAPPC to prevent the first shadow from tying up your communications lines.

When *APPC is specified, Element 2 allows you to specify whether the data in the fields of a directory entry on your system is replaced by shadowed data if the same entry also exists in the supplier system's directory.

Element 2: Replace Data

***NO:** The data in the fields of existing directory entries on your system is not replaced with data from the supplier system.

***YES:** All shadowed data is added to your system distribution directory. The data in the fields of existing directory entries on your system is replaced with shadowed data if the same entry also exists in the supplier system's directory.

***NONAPPC:** The Copy to Directory (CPYTODIR) command is used for the first shadow. It is recommended that you run the CPYTODIR command before running this command. If shadowing from the supplier system you are adding starts before CPYTODIR is run, you may lose data.

***COMPLETED:** The initial shadow has already been done using the CPYTODIR command.

RMTLOCNAME

Specifies the remote location name of the supplier system you are adding.

***SYSNAME:** The value specified on the SYSNAME parameter is used for the remote location name.

remote-location-name: Specify the full name of a remote location.

A maximum of 8 characters can be specified. The first character must be an uppercase letter A through Z, or special character $, #, or @. The name cannot contain a blank, plus sign (+), period (.), or an underscore (_). For more information, see the *APPC Programmer's Guide*.

MODE

Specifies the name of the mode that defines the sessions on the device used when shadowing data from the supplier system.

***NETATR:** The mode name specified in the network attributes is used.

mode-name: Specify the mode name.

A maximum of 8 characters can be specified. The first character must be an uppercase letter A through Z, or special character $, #, or @. The name cannot contain a blank, plus sign (+), period (.), or an underscore (_). For more information, see the *APPC Programmer's Guide*.

RMTNETID

Specifies the supplier system's remote network identifier (ID).

***LOC:** The remote network identifier (ID) associated with the remote location is used. If several remote network IDs are associated with the remote location, the system determines which remote network ID is used.

***NETATR:** The RMTNETID value specified in the system network attributes is used.

***NONE:** No remote network identifier (ID) is used.

remote-network-ID: Specify a maximum of 8 characters for the remote network ID.

The first character must be an uppercase letter A through Z, or special character $, #, or @. The name cannot contain a blank, plus sign (+), period (.), or an underscore (_). For more information, see the *APPC Programmer's Guide*.

LCLLOCNAME

Specifies the local location name. The local location name is used to identify your system to the supplier system you are adding.

***LOC:** The local location name associated with the remote location is used.

***NETATR:** The LCLLOCNAME value specified in the system network attributes is used.

local-location-name: Specify a maximum of 8 characters for the local location name.

The first character must be an uppercase letter A through Z, or special character $, #, or @. The name cannot contain a blank, plus sign (+), period (.), or an underscore (_). For more information, see the *APPC Programmer's Guide*.

TEXT

Specifies text that briefly describes the shadow system distribution directory. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SYSNAME:** The name specified on the SYSNAME parameter is used for the description.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Examples

Example 1: Shadowing a System Weekly

```
ADDDIRSHD   SYSNAME(NYCITY)
  SCD( '92/05/01' '17:00:00' )
  FRQ(*WEEKLY)  TEXT('Shadow New York System')
```

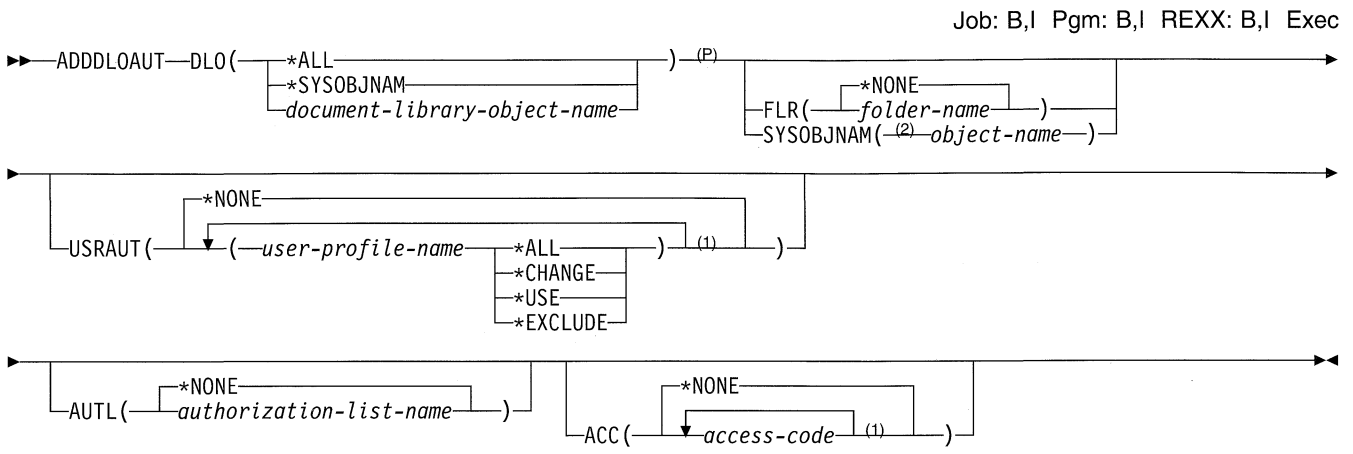
This command adds the supplier system NYCITY, which starts shadowing directory data to the local system on May 1, 1992 at 5:00 p.m. The shadow frequency is once a week. The description of the shadow system is 'Shadow New York System'.

Example 2: Shadowing a Remote System Hourly

```
ADDDIRSHD   SYSNAME(CHICAGO)
  SCD('92/04/01' '20:00:00')
  FRQ(*HOURS)  HOURS(12)
  RMTLOCNAME(CHIC01)
  LCLLOCNAME(CHICAGO1)
  TEXT('Shadow Chicago System')
```

This command adds the supplier system CHICAGO, which starts shadowing directory data to the local system on April 1, 1992 at 8:00 p.m. The frequency of shadows is every 12 hours. The description of the shadow system is 'Shadow Chicago System'. The remote location name of the CHICAGO system is CHIC01 and the local location name is CHICAGO1.

ADDDLOAUT (Add Document Library Object Authority) Command



Notes:

1 A maximum of 50 repetitions

P All parameters preceding this point can be specified in positional form.

2 The SYSOBJNAM parameter is valid only when DLO(*SYSOBJNAM) is specified.

Purpose

The Add Document Library Object Authority (ADDDLOAUT) command gives a user access to a document or folder. This command allows the user to specify authority for others in the following ways:

- Give specific authority to a user
- Give a set of users authority by specifying a previously defined authorization list
- Give a group of users access by specifying an access code

Restriction: The user of this command must be the owner of the objects, or have *ALL authority to the objects or *ALLOBJ special authority.

Required Parameter

DLO

Specifies the name of the document or folder to which authority is added.

***ALL:** All objects in the specified folder have authority added. If *ALL is specified, the FLR parameter is required.

***SYSOBJNAM:** The system object name specified in the SYSOBJNAM parameter has authority added.

document-library-object-name: Specify the user-assigned name of the document or folder. Up to 12 characters can be specified.

Optional Parameters

FLR

Specifies the name of the folder that contains the document.

***NONE:** A folder name is not specified. If DLO(document-library-object-name) is specified and the object is located in a folder, FLR(*NONE) cannot be specified. If DLO(*ALL) is specified, FLR(*NONE) cannot be specified.

folder-name: Specify the user-assigned name of the folder. The folder name can consist of a series of folder names if the object specified in the DLO parameter is located in a folder that is contained in another folder. Up to of 63 characters can be specified.

SYSOBJNAM

Specifies the system object name. This parameter is valid only when DLO(*SYSOBJNAM) or DOCL(*SYSOBJNAM) is specified. A full ten characters must be specified.

USRAUT

Specifies the name of a specific user and the user's authority.

***NONE:** No additional user authority is added.

user-profile-name: Specify the name of the user profile for whom specific authority is added.

The possible authority values are:

***ALL:** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence and specify the secu-

rity for the object, change the object, and perform basic functions on the object. The user can change ownership of the document library object.

***CHANGE:** The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority.

***USE:** The user can perform basic operations on the document library object, such as display or print document or folder content and description information. The user is prevented from changing the object. Use authority provides object operational authority and read authority.

***EXCLUDE:** The user cannot access the document library object.

AUTL

Specifies that the authority for the object named in the DLO parameter comes from the authorization list containing the list of users and their authorities.

***NONE:** An authorization list is not specified.

authorization-list-name: Specify the name of the authorization list used.

ACC

Specifies the access codes that are added. The access code must be defined to the system (using the Add Access Code (ADDACC) command) before it can be assigned to an object. An access code of zero allows public *USE authority for the object. An access code of zero cannot be added to an object if the document library object is marked personal.

***NONE:** No access code is added.

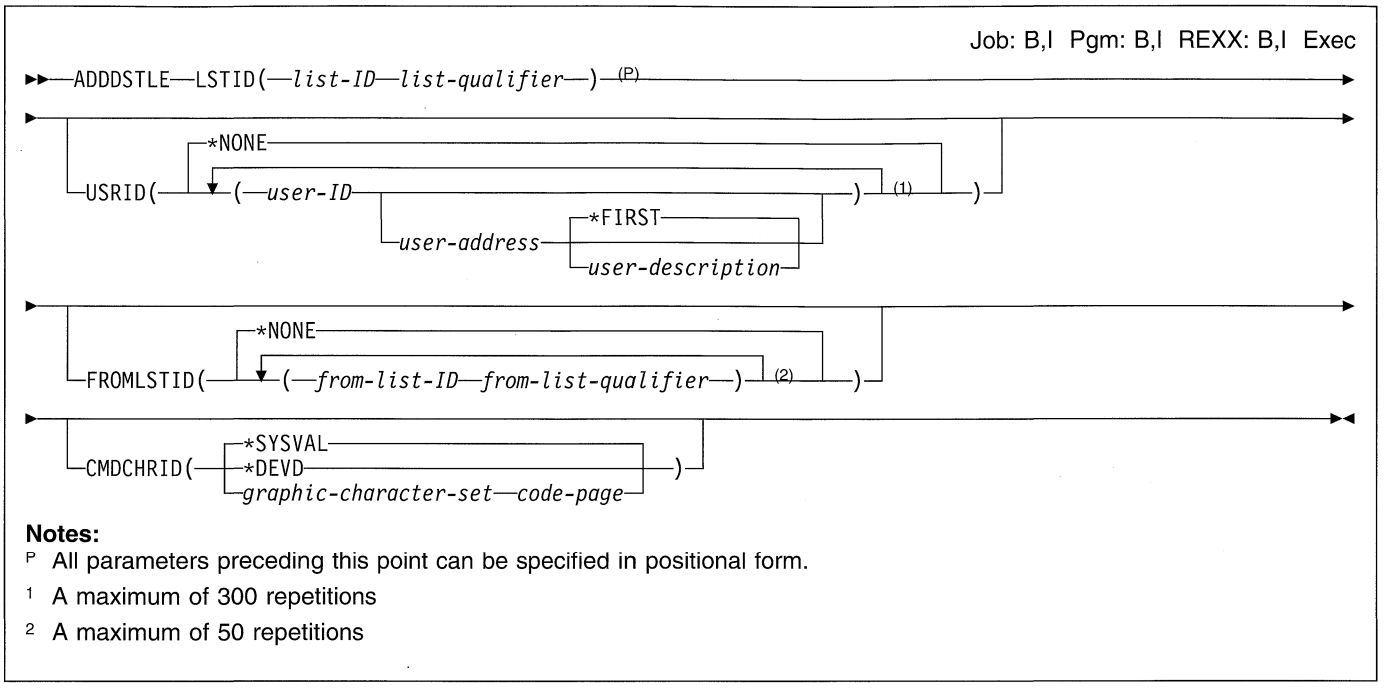
access-code: Specify the access code assigned to the object. An access code can be any number between 0 and 2047.

Example

```
ADDDLOAUT   DLO(*ALL)  USER(MIKE (*CHANGE))
  AUTL(*NONE)  FLR(MYFLR)  ACC(1023)
```

This command adds *CHANGE authority for user MIKE to all objects in the folder MYFLR. An access code of 1023 was also added to the object.

ADDDSTLE (Add Distribution List Entry) Command



Notes:

P All parameters preceding this point can be specified in positional form.

1 A maximum of 300 repetitions

2 A maximum of 50 repetitions

Purpose

The Add Distribution List Entry (ADDDSTLE) command adds new entries to an existing distribution list. A distribution list can include local, remote, indirect, and independent work station users. It can also include remote distribution lists, but not local distribution lists.

The ADDDSTLE command allows up to 300 entries to be added to a distribution list at one time. In addition, up to 50 local distribution list IDs can be specified whose members are all to become part of this list.

The distribution list must exist before this command can be run. The Create Distribution List (CRTDSTL) command can be used to create a new distribution list.

Restriction: The user of this command must have security administrator authority to add entries to a distribution list owned by someone else. Users can add entries to a distribution list they have created without restrictions.

Required Parameter

LSTID

Specifies the two-part list identifier of the distribution list that is to have entries added.

Element 1: List Identifier

list-identifier: Specify the list identifier (ID) of a distribution list.

Element 2: List Qualifier

list-qualifier: Specify the list qualifier of the distribution list.

Note: The distribution list identifier/list qualifier is in two parts, separated by at least one space. If any lowercase characters are specified, the system changes them to uppercase.

The naming rules for the two-part list ID are identical to the rules for the user ID and address. A complete description of these rules is in the *Distribution Services Network Guide*.

Optional Parameters

USRID

Specifies the user ID, address, and description of the users for whom additions to the distribution list are made. Both the user ID and address must be provided and must be separated by at least one space. If any lowercase characters are specified, the system changes and stores them as uppercase.

A list ID and address can be used in place of the user ID and address to identify a remote distribution list ID that is added to the distribution list. A remote distribution list must be defined as a remote user in the directory, or a default *ANY entry must exist.

***NONE:** No user ID or list ID is specified. Members are added to this list using the FROMLSTID parameter. If the FROMLSTID parameter also specifies *NONE, an error message is returned.

Element 1: User ID

user-ID: Specify the user ID (or the valid list ID for a remote list entry) of the user for whom the entry is made.

Element 2: User Address

user-address: Specify the user address (or the valid list address for a remote list entry) of the user for whom the entry is made. Separate the ID and address by at least one space.

Element 3: User Description

***FIRST:** The first description in the specified user ID and address (or list ID and address) is added. If only one entry exists, it is the one added to the list.

user-description: Specify the description for the user. If a list ID is specified, enter the list description. The description can be up to 50 characters in length. The description entered must be exactly the same as a directory entry or an error message is returned.

FROMLSTID

Specifies the name of an existing distribution list whose entries are added to this list. Up to 50 different list IDs can be specified. Duplicate entries are not removed. Duplicates are defined as entries that have the same user ID, address, and description.

***NONE:** No list ID is specified. Members are added to this list using the USRID parameter. If *NONE is specified for the USRID, an error message is returned.

Element 1: From List ID

from-list-ID: Specify the list ID of a distribution list whose entries are added to this list.

Element 2: From List Qualifier

from-list-qualifier: Specify the list qualifier of a distribution list whose entries are added to this list.

CMDCHRID

Specifies the character identifier (graphic character set and code page) for data being specified as parameter values on this command. This character identifier (CHRID) is related to the display device used to specify the command. More information about CHRID processing is in the *Guide to Programming Displays*.

***SYSVAL:** The system determines the graphic character set and code page values for the command parameters from the QCHRID system values.

***DEVD:** The system determines the graphic character set and code page values for the command parameter from the display device description where the command is entered. This option is valid only when specified from an interactive job. If this value is specified in an interactive CL program or a batch job, an error message is sent.

Element 1: Character Set

graphic-character-set: Specify the character set used to create the command parameters. Valid values range from 1 through 9999 characters.

Element 2: Code Page

code-page: Specify the code page. Valid values range from 1 through 9999.

Example

```
ADDDSTLE LSTID(CHICAGO DLIST)
  USRID((HURST NEWYORK 'Manager of Payroll')
  (LEE DEPT554 *FIRST)
  (BOCA DLIST 'Remote Distribution
  list for Boca') (ERIC WAREHSE))
  FROMLSTID((DEPT48K DLIST) (ALLMGRS DLIST))
```

This command specifies that four user IDs are added to the distribution list CHICAGO DLIST. The third user ID is in fact a remote distribution list. The fourth user ID (ERIC WAREHSE) defaults to the first description for that user ID. In addition, all of the entries in two distribution lists are added to this distribution list.

Additional Considerations

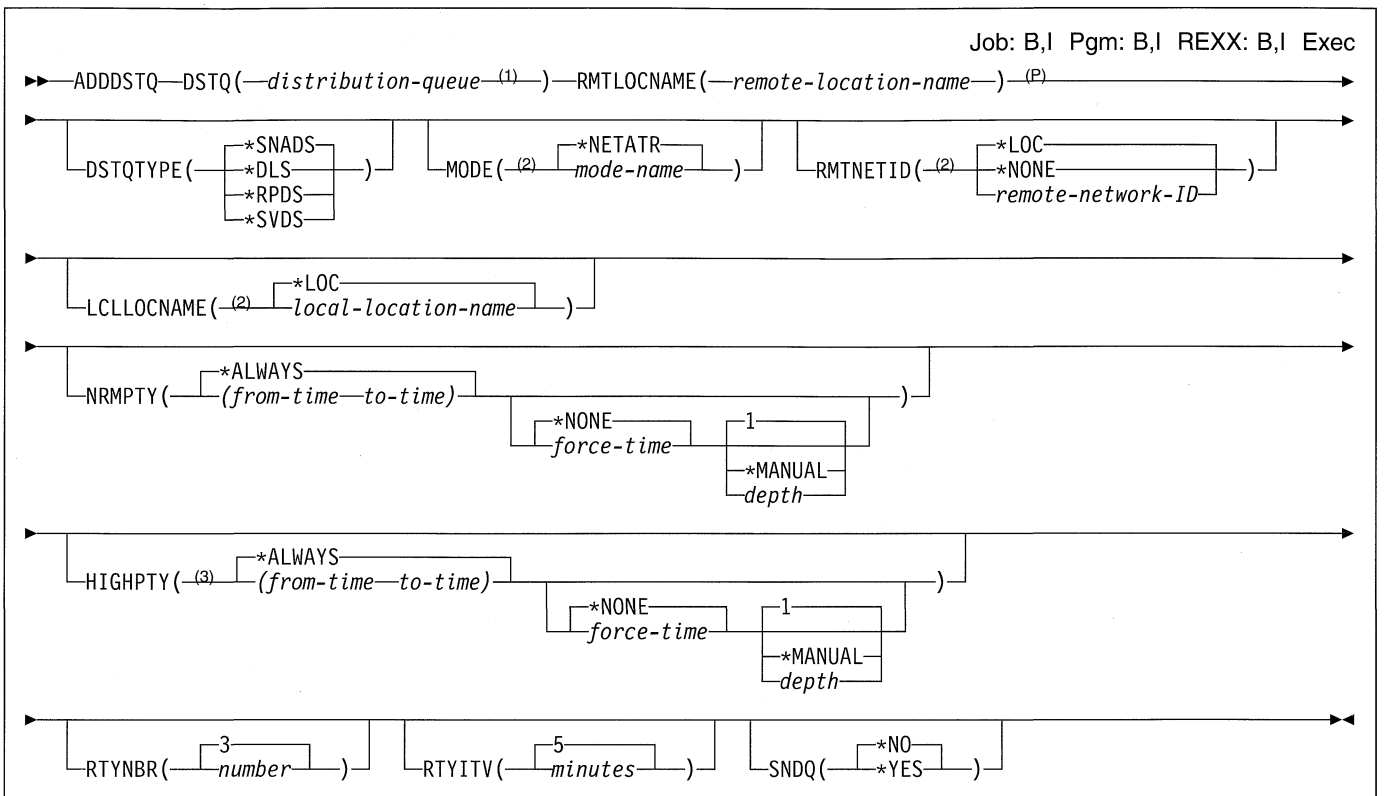
The list ID specified to have entries added to it must already exist in the directory. If the list does not exist, an error message is returned. The Create Distribution List (CRTDSTL) command is used to create a new distribution list.

Up to 300 sets of user IDs, addresses and user descriptions can be entered on this command. Each set of user ID and address is examined to determine whether it exists in the directory. If it is not in the directory, the directory is searched for an entry with user ID *ANY and an address that matches the address of the specified user. If this is found, the user ID and address is added to the distribution list. If it is not found, but a *ANY *ANY entry exists, the user ID is added to the list. Otherwise an error message is returned, stating that the user ID is not valid.

The command processes each set of user ID, address, and description. Each invalid set returns an error message, but the valid sets are added to the distribution list. At the end of processing of the command, a final message is returned indicating how many entries are in fact added to the distribution list, and how many entries are invalid.

The command allows input of up to 50 distribution list IDs whose members are added to the specified distribution list. If any list ID is not found in the directory, an error message is returned. Like the sets of user IDs, and addresses, any valid list IDs will have their members added to the distribution list, and a message is returned for each invalid list ID.

ADDDSTQ (Add Distribution Queue) Command



Notes:

1 The distribution queue name cannot be only blanks.

2 This parameter is not valid when DSTQTYPE(*RPDS) is specified.

3 This parameter is not valid when DSTQTYPE(*SVDS) is specified.

P All parameters preceding this point can be specified in positional form.

Purpose

The Add Distribution Queue (ADDDSTQ) command adds an entry to the distribution services queue table. Distribution queues are used to store distributions before they are sent or forwarded to other systems.

Interactive display support is provided by the Configure Distribution Services (CFGDSTSRV) command. More information about configuring a distribution network is in the *Distribution Services Network Guide*.

Distribution queue names are translated to the graphic character set and code page 930 500, using the job's coded character set identifier (CCSID).

Restrictions:

1. This command is shipped with public *EXCLUDE authority, and the QPGMR and QSYSOPR user profiles have private authorities to use the command.
2. The combination of remote location name, mode, remote network identifier, and local location name must be unique within the type of distribution queue. This combination does not need to be unique within the system, for SNA distribution services (SNADS) distribution queues in the distribution services queue table (SNADS-type distribution queues), and for SystemView* distribution services (SVDS) distribution queues (SVDS-type distribution queues). The default value *LOC, which can be specified on the RMTNETID parameter and the LCLLOCNAME parameter, and the default value *NETATR, which can be specified on the MODE parameter, represent any possible values that the system determines are not already configured for another SNADS or SVDS distribution queue of the same type.
3. A unique remote location name must be specified for each RPDS-type distribution queue in the queue table. RPDS queues do not use modes, remote network identifiers, or local location names.
4. Configuration in the routing table is not required for SVDS-type distribution queues. SVDS queues may be configured optionally in the SNADS routing table. However, normal SNADS mail can neither be routed to change management queues nor be received through

change management connections, and change management connections can neither be routed to SNADS queues nor be received through SNADS connections.

5. SVDS-type distribution queues can support only a single queue view (the queue is not divided into normal and priority halves). For configurations and operations purposes, only the normal queue is specified.
6. Messages that report errors about distribution queues may display or print different characters than the user entered for the distribution queue name because of internal system transformations. Similarly (depending on the language used for the work station), the internal value for a distribution queue name may differ from the characters shown on the Work with Distribution Queue (WRKDSTQ) command. An error may be reported if the character-string value specified for the DSTQ parameter does not match the rules for an internal distribution queue value or if it does not match the internal value for any defined distribution queue (ignoring case differences).

Required Parameters

DSTQ

Specifies a maximum of 16 characters for the name of the distribution queue being added to the distribution services queue table.

RMTLOCNAME

Specifies the name of the remote location where distributions are sent from this distribution queue. The remote location name must be configured in the device description of the device used when sending distributions to another system from this distribution queue.

Optional Parameters

DSTQTYPE

Specifies the type of distribution queue being added.

***SNADS:** SNADS is the distribution queue type. SNADS queues are used to send distributions within a SNADS network.

***DLS:** Document library services (DLS) is the distribution queue type. DLS queues are used to communicate between the local system and document library services on a remote system.

***RPDS:** RPDS is the distribution queue type. RPDS queues are used to communicate between the local system and the AS/400 VM/MVS bridge, or JES (2,3) on System/370*-type systems, and for the SNADS extended bridge function of the AS/400 Communications Utilities licensed program.

***SVDS:** SystemView distribution services (SVDS) is the distribution queue type. SVDS queues support the communications bridge between a SNADS network and SAA* SystemView System Manager/400 change management. An SVDS queue must be defined in order to receive from as well as send to a remote system using change management.

MODE

Specifies the name of the mode that defines the sessions on the device used by the distribution queue. This parameter is ignored if *RPDS is specified on the DSTQTYPE parameter.

***NETATR:** The mode name specified in the network attributes is used.

mode-name: Specify a maximum of 8 characters for the name of the mode. The mode name cannot be CPSVCMG or SNASVCMG; these mode names are reserved for system use.

RMTNETID

Specifies the remote network identifier of the remote network to which this distribution queue sends distributions. This parameter is ignored if *RPDS is specified on the DSTQTYPE parameter.

***LOC:** The remote network identifier defined in the device description used by this distribution queue is used.

***NONE:** No remote network identifier is specified.

remote-network-ID: Specify the remote network identifier.

LCLLOCNAME

Specifies the name used to identify the local system to remote systems in the network. It is recommended that the name be the same as the local system name. This parameter is ignored if *RPDS is specified on the DSTQTYPE parameter.

***LOC:** The local location name defined in the device description used by this distribution queue is used.

local-location-name: Specify a maximum of 8 characters for the local location name.

NRMPTY

Specifies the queue sending conditions for distributions having a service level of data low.

Element 1: Send Time

The send time is the time period during which queued distributions of this priority are sent from this distribution queue. If no time period is specified, the transmissions are controlled by queue depth and are not related to time. The time must be specified in the 24-hour format hhmm where h = hours and m = minutes.

***ALWAYS:** Distributions of this priority are sent from this distribution queue regardless of the time of day.

from-time: Specify the time of day at which the system starts sending distributions of this priority from this distribution queue if the value specified for send depth is reached. If *from-time* is specified, *to-time* must also be specified.

to-time: Specify the time of day at which the system stops sending distributions of this priority from this distribution queue. If *to-time* is specified, *from-time* must also be specified.

Element 2: Force Time

Force time is a specific time during which distributions of this priority are sent regardless of queue depth. If *ALWAYS is specified for the send time, the force time can be set to any time of day. If a specific to-time and from-time is specified for the send time, the force time must occur within that time period. The time must be specified in the 24-hour format hhmm where h = hours and m = minutes.

***NONE:** No force time is specified.

force-time: Specify the force time.

Element 3: Depth

Queue depth specifies the number of distributions of this priority that are on the queue before sending begins.

1: Distributions are sent when they are put on the queue.

***MANUAL:** Distributions are sent only when an operator manually sends them using the Work with Distribution Queue (WRKDSTQ) command or the Send Distribution Queue (SNDDSTQ) command.

depth: Specify the number of distributions of this priority that are on this distribution queue before any are sent. Valid values range from 1 through 999.

HIGHPTY

Specifies the queue sending conditions for distributions having a service level of fast, status, or data high. On this parameter, time must be specified in the 24-hour format hhmm where h = hours and m = minutes.

Element 1: Send Time

***ALWAYS:** Distributions of this priority are sent from this distribution queue regardless of the time of day.

from-time: Specify the time of day at which the system starts sending distributions of this priority from this distribution queue if the value specified for send depth is reached. If *from-time* is specified, *to-time* must also be specified.

to-time: Specify the time of day at which the system stops sending distributions of this priority from this distribution queue. If *to-time* is specified, *from-time* must also be specified.

Element 2: Force Time

***NONE:** No force time is specified.

force-time: Specify the force time.

Element 3: Depth

1: Distributions are sent when they are put on the queue.

***MANUAL:** Distributions are sent only when an operator manually sends them using the WRKDSTQ command or the SNDDSTQ command.

depth: Specify the number of distributions of this priority that are on this distribution queue before any are sent. Valid values range from 1 through 999.

RTYNBR

Specifies the maximum number of times the system attempts to resend distributions from this distribution queue after a failure occurs. This parameter applies to communications line failures and recoverable distribution failures on a remote system. The SNADS job serving this distribution queue ends when the number of retries is exceeded.

3: The system attempts to resend distributions a maximum of 3 times after a failure.

number: Specify the maximum number of times the system can attempt to resend distributions after a failure. Valid values range from 0 through 9999.

RTYITV

Specifies the interval (in minutes) between each retry attempt.

5: The number of minutes between retries is 5.

minutes: Specify the interval between retries. Valid values range from 0 through 9999 minutes.

SNDQ

Specifies whether this distribution queue ignores the send time and depth values specified on the NRMPTY and HIGHPTY parameters and begins sending when a distribution is received from the SNADS system to which the queue sends its distributions.

This parameter is valid only if *SNADS is specified on the DSTQTYPE parameter.

***NO:** Distributions are sent from this queue only when the queue's sending conditions are met.

***YES:** This distribution queue begins sending when distributions are received from the SNADS system to which the queue sends its distributions, regardless of the queue's other sending conditions. Distributions are automatically sent for manual queues (queues that have no specified depth variable).

Examples

Example 1: Adding a SNADS Distribution Queue

```
ADDDSTQ   DSTQ(CHICAGO)  RMTLOCNAME(CHICAGOLU)
  MODE(NEWMODE)
```

This command adds a distribution queue named CHICAGO. The queue uses remote location name CHICAGOLU and mode NEWMODE when sending SNADS distributions.

Example 2: Adding a DLS Distribution Queue

```
ADDDSTQ   DSTQ(DLSQUEUE)  DSTQTYPE(*DLS)
  RMTLOCNAME(DLSLU)  MODE(DLSMODE)
```

This command adds a DLS type of distribution queue named DLSQUEUE. The queue uses remote location name DLSLU and mode DLSMODE when sending DLS requests.

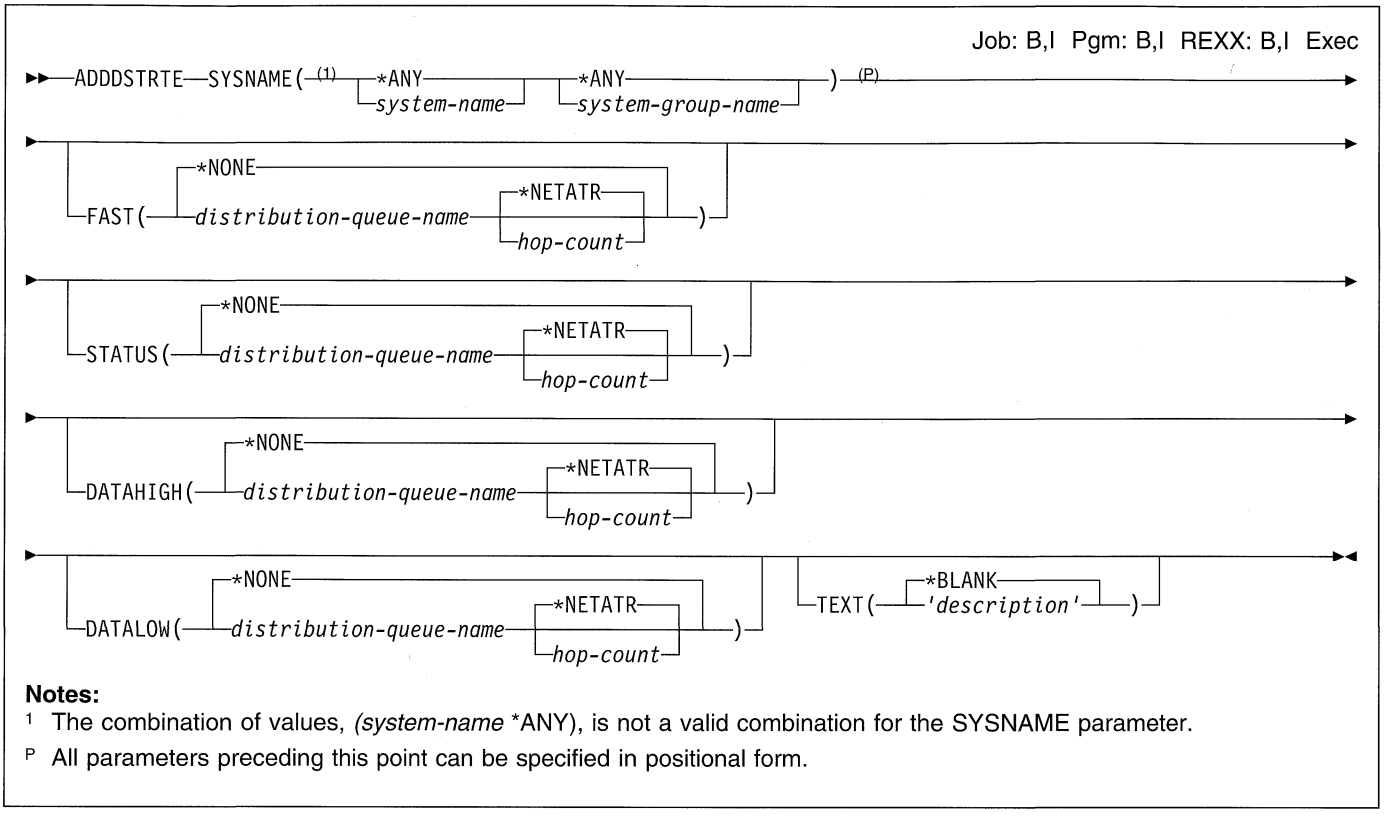
Example 3: Adding an SVDS Distribution Queue

```
ADDDSTQ   CHICACM  CHIGAGOLU  DSTQTYPE(*SVDS)
```

This command adds an SVDS type of distribution queue named CHICACM. The queue uses remote location name CHICAGOLU when sending and receiving SVDS change management distributions.

Note: In this example, the distribution queue name and remote location name parameters are specified by position.

ADDDSTRTE (Add Distribution Route) Command



Notes:

1 The combination of values, *(system-name* *ANY), is not a valid combination for the SYSNAME parameter.

P All parameters preceding this point can be specified in positional form.

Purpose

The Add Distribution Route (ADDDSTRTE) command adds an entry to the distribution services routing table. The routing table determines which distribution queue receives a distribution on its way to a particular destination.

Distributions are routed to distribution queues based on service levels. One or more service levels must be specified for each routing table entry. This system will not route distributions for service levels that have not been configured. Normally, all service levels routed to the same destination use the same distribution queue. However, the user can configure several distribution queues for one destination based on distribution service levels.

Interactive display support is provided by the Configure Distribution Services (CFGDSTSRV) command.

System names, system group names, and distribution queue names are translated to the graphic character set and code page 930 500, using the job's coded character set identifier (CCSID).

Restrictions:

1. This command is shipped with public *EXCLUDE authority, and the QPGMR and QSYSOPR user profiles have private authorities to use the command.
2. An error occurs if a distribution route specifying a SystemView distribution services (SVDS) type of distribution queue includes another type of distribution queue (such as SNA distribution services (SNADS) or VM/MVS bridge (RPDS)).

Required Parameter

SYSNAME

Specifies the system name and group name of the remote system that is the destination for this routing table entry. The local system name cannot be specified unless a group name is also specified.

Element 1: System Name

***ANY:** *ANY is used for the system name. When *ANY and a system group name are specified, any system in the group is the destination for the routing table entry. Only one *ANY value is allowed for each group in the routing table.

system-name: Specify a maximum of 8 characters for the name of the remote system that is the destination for this routing table entry.

Element 2: System Group Name

***ANY:** *ANY is used for the system group name. *ANY can be specified for the group name only if *ANY is also specified for the system name. Only one SYSNAME(*ANY *ANY) entry is allowed in the routing table and is used to resolve a distribution destination that does not match any other routing table entries.

system-group-name: Specify a maximum of 8 characters for the system group name. The system name and group name must be separated by at least one blank.

Optional Parameters

FAST

Specifies the distribution queue and maximum hop count to the destination system for fast service level distributions. The fast service level is the highest priority level.

The maximum hop count is the maximum number of times in a SNADS network that a distribution can be routed back and forth (hop) between the systems participating in the SNADS level routing, including the hop to the final destination system. The maximum hop count does not include the hops made by advanced peer-to-peer networking (APPN) routing. If the maximum number of hops is exceeded, the distribution is ended and an error is sent to the user who originally sent the distribution.

***NONE:** No distribution queue is specified for distributions requiring a fast service level. Distributions requiring fast service cannot be routed using this routing table entry.

Element 1: Distribution Queue

distribution-queue-name: Specify the name of the distribution queue to which fast distributions using this routing entry are sent. The distribution queue must already exist and cannot be a DLS (document library services) type of queue.

Element 2: Maximum Hop Count

***NETATR:** The system network attribute value for the maximum hop count is used. The current system default value can be displayed using the Display Network Attributes (DSPNETA) command.

hop-count: Specify the maximum hop count. Valid values range from 1 through 255.

STATUS

Specifies the distribution queue and maximum hop count to the destination system for status service level distributions. The status service level is used for network status and other feedback information.

***NONE:** No distribution queue is specified for distributions requiring a status service level. Distributions requiring status service cannot be routed using this routing table entry.

Element 1: Distribution Queue

distribution-queue-name: Specify the name of the distribution queue to which status distributions using this routing entry are sent. The distribution queue must already exist and cannot be a DLS-type queue.

Element 2: Maximum Hop Count

***NETATR:** The system network attribute value for the maximum hop count is used. The current system value can be displayed using the Display Network Attributes (DSPNETA) command.

hop-count: Specify the maximum hop count. Valid values range from 1 through 255.

DATAHIGH

Specifies the distribution queue and maximum hop count to the destination system for data high service level distributions. The data high service level is used for high priority data traffic.

***NONE:** No distribution queue is specified for distributions requiring a data high service level. Distributions requiring data high service cannot be routed using this routing table entry.

Element 1: Distribution Queue

distribution-queue-name: Specify the name of the distribution queue to which data high distributions using this routing entry are sent. The distribution queue must already exist and cannot be a DLS-type queue.

Element 2: Maximum Hop Count

***NETATR:** The system network attribute value for the maximum hop count is used. The current system value can be displayed using the Display Network Attributes (DSPNETA) command.

hop-count: Specify the maximum hop count. Valid values range from 1 through 255.

DATALOW

Specifies the distribution queue and maximum hop count to the destination system for data low service level distributions. The data low service level is used for most data traffic.

***NONE:** No distribution queue is specified for distributions requiring a data low service level. Distributions requiring data low service cannot be routed using this routing table entry.

Element 1: Distribution Queue

distribution-queue-name: Specify the name of the distribution queue to which data low distributions using this routing entry are sent. The distribution queue must already exist and cannot be a DLS-type queue.

Element 2: Maximum Hop Count

***NETATR:** The system network attribute value for the maximum hop count is used. The current system value can be displayed using the Display Network Attributes (DSPNETA) command.

hop-count: Specify the maximum hop count. Valid values range from 1 through 255.

TEXT

Specifies text that briefly describes the distribution route. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

Note: Double-byte character set (DBCS) characters can be entered on this parameter.

***BLANK:** No text is associated with the new distribution route.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Examples

Example 1: Adding an Entry for an Adjacent System

```
ADDDSTRTE   SYSNAME(SYSTEMA GROUPA)
  FAST(SYSTEMA)  STATUS(SYSTEMA)
  DATAHIGH(SYSTEMA)  DATALOW(SYSTEMA)
```

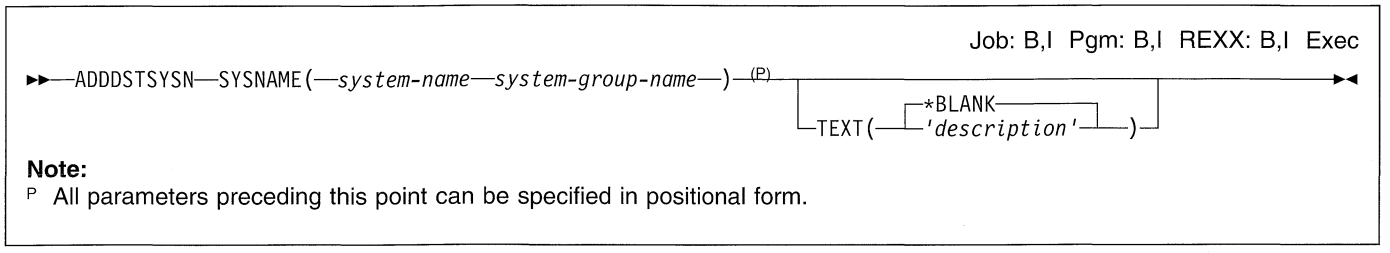
This command adds a routing table entry for a system that is directly connected to this system (via a physical advanced program-to-program communications (APPC) connection or a logical APPN connection). The distribution queue is given the same name as the destination system. The hop count defaults to the system default value.

Example 2: Adding a Generic Routing Table Entry

```
ADDDSTRTE   SYSNAME(*ANY GROUPNM1)
  FAST(SYSTEMA)  STATUS(SYSTEMA)
  DATAHIGH(SYSTEMA)  DATALOW(SYSTEMA)
```

This command adds a routing table entry for all systems in system group GROUPNM1. The distribution queue SYSTEMA is used to route distributions to all systems in the group.

ADDDSTSYSN (Add Distribution Secondary System Name) Command



Note:
P All parameters preceding this point can be specified in positional form.

Purpose

The Add Distribution Secondary System Name (ADDDSTSYSN) command adds an entry to the distribution services secondary system name table. The table contains names of all of the alternate (or alias) system names for which the local system receives and may redirect distributions. The SNA distribution services (SNADS) function automatically receives distributions with the local system as the destination system name, so the local system cannot be added to the secondary system name table.

This command can be used to:

- Give the local system a group name
- Rename the local system
- Combine two systems into one system or divide one system into two systems
- Allow a system to receive mail destined for a system from which many users have moved

Interactive display support is provided by the Configure Distribution Services (CFGDSTSRV) command.

System names and system group names are translated to the graphic character set and code page 930 500, using the job's coded character set identifier (CCSID).

Restrictions:

1. This command is shipped with public *EXCLUDE authority, and the QPGMR and QSYSOPR user profiles have private authorities to the command.
2. The secondary system name table does not operate with SystemView distribution services (SVDS) types of distributions.

Required Parameter

SYSNAME
Specifies the alternate system name and system group name being added to the distribution services secondary system name table. The current system name cannot be specified unless a group name is also specified.

Element 1: System Name

system-name: Specify a maximum of 8 characters for the name of the system for which the local system is to receive distributions.

Element 2: System Group Name

system-group-name: Specify a maximum of 8 characters for the system group name of the system for which the local system is to receive distributions. The system name and group name must be separated by at least one blank.

Optional Parameter

TEXT
Specifies text that briefly describes the secondary system name. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

Note: Double-byte character set (DBCS) characters can be entered on this parameter.

***BLANK:** No text is associated with the secondary system name.

'description': Specify a maximum of 50 characters of text, enclosed in apostrophes.

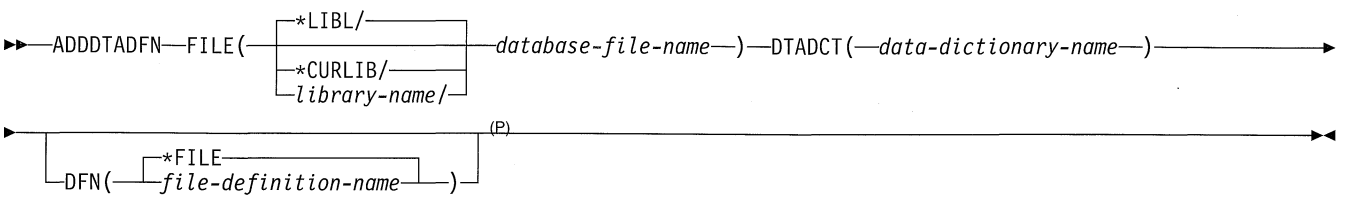
Example

```
ADDDSTSYSN   SYSNAME(SYS2LAJ1 ROCHESTR)
```

This command adds the system named SYS2LAJ1 ROCHESTR to the distribution services secondary system name table. The local system will receive distributions that contain SYS2LAJ1 ROCHESTR as the destination system name.

If the local system is named SYS2LAJ1, this command allows the local system to participate in a network that requires a group name of ROCHESTR for each participating system.

ADDDTADFN (Add Data Definition) Command



Note:
P All parameters preceding this point can be specified in positional form.

Purpose

The Add Data Definition (ADDDTADFN) command allows the user to copy file, record format, and field definitions from an externally described database file to a data dictionary.

When definitions are added to a dictionary, the system does a search to find out if the dictionary contains a definition with the same name as the one being added. If an exact match of the definition is found, the existing definition is used. If an exact match is not found, a new version of the definition is created.

Database files using the following functions are not added to a dictionary:

- Access path sharing
- Alternate collating sequence
- Program-described file
- Join logical file
- Logical file
  -with select/omit specifications
  -based on more than one file

Only format and field definitions of database files using the following functions are added to a dictionary:

- Field default values
- Field validity check codes
- Key fields defined using names based on physical files
- Derived fields

Note: When adding a file that is already linked, the current link is ended and then the definition is added and linked.

Required Parameters

FILE
Specifies the qualified name of the externally described database file from which the data definition is copied to the specified dictionary. A database file name must be specified.

The name of the file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

database-file-name: Specify the name of the database file from which the definition is copied to the specified dictionary.

DTADCT
Specifies the name of the dictionary to which the definitions are added.

Optional Parameter

DFN
Specifies the name given to the file definition when it is copied into the dictionary.

***FILE:** The name of the file definition is the same as the database file specified in the FILE parameter.

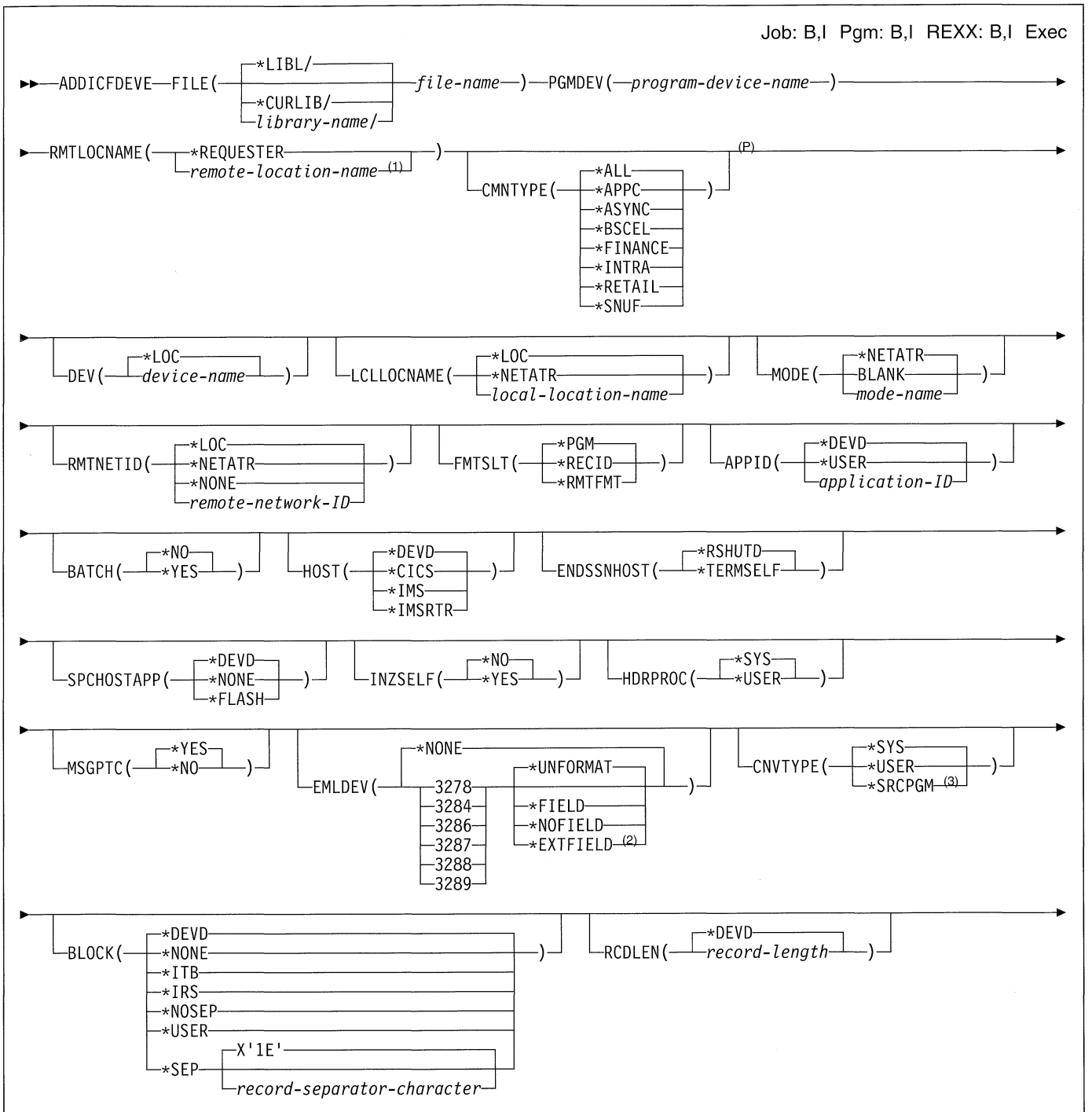
file-definition-name: Specify the name given to the file definition when copied to the data dictionary.

Example

```
ADDDTADFN   FILE(MYLIB/MYFILE)
  DTADCT(MINE)  DFN(*FILE)
```

This command copies the definitions from MYFILE located in library MYLIB to the dictionary MINE. The file definition has the same name as the database file specified in the FILE parameter.

ADDICFDEVE (Add Intersystem Communications Function Program Device Entry) Command



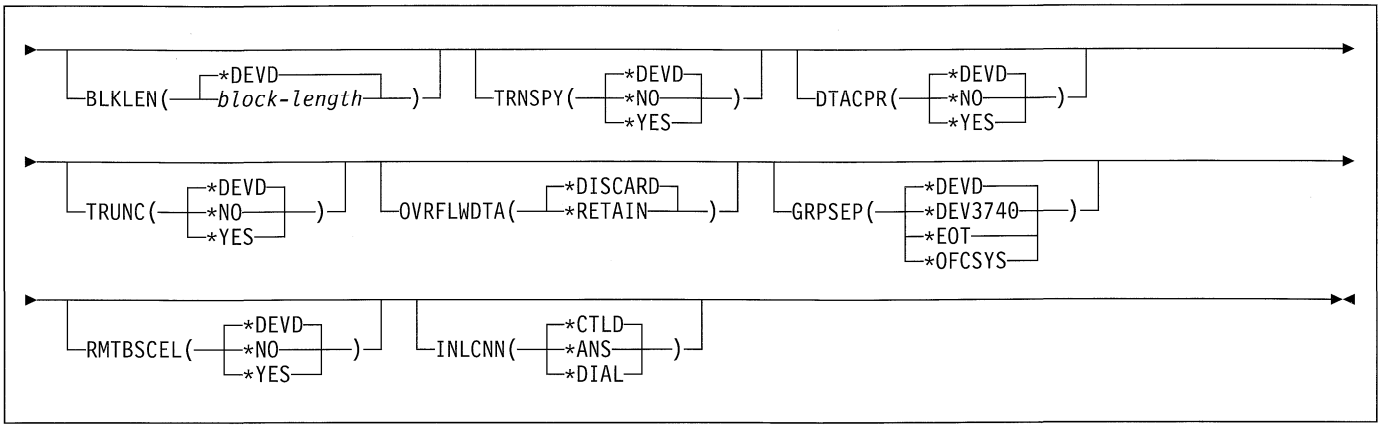
Notes:

1 This value cannot be specified with CNVTYPE (*SRCPGM).

P All parameters preceding this point can be specified in positional form.

2 The value *EXTFIELD is valid only when 3278 is specified.

3 *SRCPGM or CNVTYPE(*SRCPGM) can only be specified with RMTLOCNAME (*REQUESTER).



Purpose

The Add Intersystem Communications Function Program Device Entry (ADDICFDEVE) command adds a program device entry with the specified name and attributes to an Intersystem Communications Function (ICF) file.

Required Parameters

FILE

Specifies the qualified name of the ICF file to which the ICF program device entry is added.

The name of the ICF file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

file-name: Specify the name of the file that contains the program device entry to be added.

PGMDEV

Specifies the name by which the ICF program device entry being added is known. The total number of program device entries that can be added (specified on the ADDICFDEVE or CHGICFDEVE command) to an ICF file is determined by the MAXPGMDEV parameter on the Create Intersystem Communications Function File (CRTICFF) command or the Change Intersystem Communications Function File (CHGICFF) command.

The name specified for this parameter is the ICF program device entry with which the user's program communicates. This name is used on device-specific input/output operations to identify the program device and its attributes. Although the user may specify the same remote location name on more than one device entry, each program device name must be unique among the entries for the ICF file. This allows the user to have more than one session to the same remote location, and/or to have different attribute values for each session to the same remote location.

Note: Refer to the *APPC Programmer's Guide* for information on how the system uses the RMTLOCNAME, DEV, LCLLOCNAME, and RMTNETID parameters to select an APPC device description.

RMTLOCNAME

Specifies the remote location name of the system with which this object communicates.

***REQUESTER:** The name used to refer to the communications device through which the program is started is used. The session that is assigned when the program device is acquired is the same session in which the program start request is received. If the program is not started as a result of a program start request, the acquire operation of the program device fails. The target program uses *REQUESTER as the remote location name in the intersystem communications function (ICF) file to connect to the session that the source program used to send the program start request.

The *REQUESTER value can be specified on only one program device entry and is valid only for a target communication job. If *REQUESTER is specified in any other type of job, a message is sent.

remote-location-name: Specify the full name of a remote location. The remote location does not need to exist at the time this command is run, but it must exist (be configured on the system as a device description or in the advanced peer-to-peer network (APPN)) for this remote location at the time the program acquires the program device. The same remote location name may be specified for many different program device entries. However, only one program device name associated with each asynchronous (ASYNC), SNA upline facility (SNUF), or binary synchronous communication equivalence link (BSCEL) remote location may be added to the file at any one time. This value cannot be specified with CNVTYPE(*SRCPGM).

Optional Parameters

CMNTYPE

Specifies which type of communications parameters are shown on the prompt display. This parameter is used only for the purpose of prompting. The value specified for this parameter determines the subset of other parameters that are shown (prompted) for the user.

***ALL:** The parameters for all of the communications types appear in the prompt.

***APPC:** The advanced program-to-program communications (APPC) parameters appear in the prompt.

***ASYNC:** The asynchronous (ASYNC) parameters appear in the prompt.

***BSCEL:** The binary synchronous communications equivalence link (BSCEL) parameters appear in the prompt.

***FINANCE:** The finance parameters appear in the prompt.

***INTRA:** The intrasystem (INTRA) parameters appear in the prompt.

***RETAIL:** The retail parameters appear in the prompt.

***SNUF:** The SNA upline facility (SNUF) parameters appear in the prompt.

DEV

Specifies the communications device used in the remote location. This parameter should only be specified for APPC, FINANCE, RETAIL, INTRA, and SNUF communications types.

***LOC:** The device associated with the remote location is used. If several devices are associated with the remote location, the system determines which device is used.

device-name: Specify the name of a communications device associated with the remote location. If the device name is not valid for the remote location, a message is sent when the program device entry is acquired. More information on device names is in the *APPC Programmer's Guide*.

LCLLOCNAME

Specifies the local location name.

This parameter applies only to the APPC communications type and is ignored for all other communications types.

***LOC:** The device associated with the remote location is used. If several devices are associated with the remote location, the system determines which device is used.

***NETATR:** The LCLLOCNAME value specified in the system network attributes is used.

local-location-name: Specify the local location name to be associated with the program device entry. If the local location name is not valid for the remote location or remote location and device, an escape message is sent when the program device entry is acquired.

MODE

Specifies the mode name used. This parameter applies only to the APPC communications type and is ignored for all other communications types.

***NETATR:** The mode name specified in the network attributes is used.

***BLANK:** The mode name consisting of 8 blank characters is used.

mode-name: Specify a mode name for the APPC communications device. If the specified mode is not valid for any combination of remote location, device, local location, and remote network ID, an escape message is sent when the program device entry is acquired.

RMTNETID

Specifies the remote network ID used with the remote location. This parameter applies to the APPC communications type only and is ignored for all other communications types.

***LOC:** The remote network identifier (ID) associated with the remote location is used. If several remote network IDs are associated with the remote location, the system determines which remote network ID is used.

***NETATR:** The RMTNETID value specified in the system network attributes is used.

***NONE:** No remote network identifier (ID) is used.

remote-network-ID: Specify a remote network ID for the APPC communications device.

FMTSLT

Specifies the record format selection used for input operations.

***PGM:** The program determines which record formats are selected. If an input (read) operation with a record format name is specified, that format is selected. If an input operation without a record format is specified, the default format (the first record format in the file) is selected. This also means that if there are any record identification (RECID) parameters specified in the data description specifications (DDS) for the file, or if any remote formats are received, they are not taken into consideration when the record is selected.

***RECID:** The record identification (RECID) keywords specified in the DDS for the file are used to do record selection. If there are no RECID keywords in the file, an error message is sent, the acquire operation of the program device ends, and the device is not acquired.

***RMTFMT:** The remote format names received from the sending system are used for record selection. If the device is not an APPC or INTRA device and *RMTFMT is specified when the program device entry is acquired, a run time error message is sent.

APPID

Specifies (in characters) the virtual telecommunications access method (VTAM*) identifier of the Customer Information Control System for Virtual Storage (CICS*/VS) or Information Management System/Virtual Storage (IMS/VS) host subsystem sent with the sign-on message. This parameter applies to the SNUF and Finance communications type only and is ignored for all other communications types.

***DEVD:** The application identifier specified in the device description is sent with the sign-on message.

***USER:** The application program can send messages or a logon to the host. This is valid only when using the 3270 program interface.

application-ID: Specify the application identifier that is sent with the sign-on message.

BATCH

Specifies, for both CICS/VS and IMS/VS, whether this session is to be used for batch jobs. This parameter applies to the SNUF, RETAIL and INTRA communications types only and is ignored for all other communications types.

***NO:** Batch jobs do not occur.

***YES:** Batch jobs occur.

HOST

Specifies the host or remote subsystem with which this session communicates. This parameter applies to the SNUF communications types only and is ignored for all other communications types.

***DEVD:** The host system specified in the device description is used.

***CICS:** The session communicates with CICS/VS.

***IMS:** The session communicates with IMS/VS.

***IMSRTR:** The session communicates with IMS/VS using the ready-to-receive option.

ENDSSNHOST

Specifies how SNUF ends the session with the host. This parameter is only valid for SNUF communications.

***RSHUTD:** SNUF sends a request-shut-down command to the host.

***TERMSELF:** SNUF sends a terminate-self command to the host. This value may be required if the value *RSHUTD fails to end a session with a non-IBM host.

SPCHOSTAPP

Specifies whether SNUF customizes support for special host applications outside the CICS or IMS application layer.

***DEVD:** The special host application specified in the device description is used.

***NONE:** SNUF does not customize support for special host applications.

***FLASH:** SNUF customizes support for the Federal Link Access for Secondary Half-sessions (*FLASH) protocol application.

INZSELF

Specifies whether a formatted INIT-SELF is built in place of the unformatted sign-on normally sent by SNUF to the host.

***NO:** The unformatted default sign-on provided by SNUF is used.

***YES:** The formatted INIT-SELF provided by SNUF is used.

HDRPROC

Specifies, for both CICS/VS and IMS/VS, whether the received function management headers are passed to the application program. This parameter applies to the SNUF communication type only and is ignored for all other communications types.

***SYS:** SNA upline facility (SNUF) removes function management headers before passing data to the program.

***USER:** Function management headers are passed with the data to the program.

MSGPTC

Specifies, for both CICS/VS and IMS/VS, whether message protection is used for this session. This parameter applies to the SNUF communications types only and is ignored for all other communications types.

***YES:** Message protection is used. SNUF saves messages until they are responded to and tries synchronization again if errors occur. *YES is only valid when BATCH(*NO) is also specified.

***NO:** Message protection is not used.

EMLDEV

Specifies that the program device entry is used to send and receive data streams to and from specific types of 3270 display or printer devices being emulated. This parameter consists of an emulation device type and an emulation device data format. The emulation device data format specifies the format of the type 3270 data stream being sent or received. A 20- or 32-byte common header that contains type 3270 command and data flow information is located at the start of the I/O buffer that is sending or receiving the type 3270 data stream. This parameter applies to the SNUF communications type only. This parameter can be specified as a list of two values (elements) or as a single value (*NONE).

***NONE:** This program device entry is not used for sending and receiving 3270 data streams.

Element 1: Type of Device

3278: The data stream is for a 3279, 3278 or 3277 display device.

3284: The data stream is for a 3284 Printer.

3286: The data stream is for a 3286 Printer.

3287: The data stream is for a 3287 Printer.

3288: The data stream is for a 3288 Printer.

3289: The data stream is for a 3289 Printer.

Element 2: Format of Data Stream

***UNFORMAT:** An unformatted 3270 data stream is sent or received. The user application program must translate the data stream into a display or printer image.

***FIELD:** A formatted 3270 data stream is sent or received. The formatted 3270 data stream contains a display or printer image followed by field definitions. The field definitions indicate the location and characteristics of each field.

***NOFIELD:** A formatted 3270 data stream that has no field definitions but contains a display or printer image is sent or received.

***EXTFIELD:** A formatted 3270 data stream contains extended field attribute information. The extended field attribute information is in the field definitions which follow the display image. The field definitions indicate the location and characteristics of each field. The value *EXTFIELD is valid only if the value 3278 is specified for the type of device on the EMLDEV parameter.

Note: If *FIELD, *NOFIELD, or *EXTFIELD is specified, BATCH(*NO) must also be specified.

CNVTYPE

Specifies the conversation type for which the application program is designed. This parameter applies to the APPC communication type only and is ignored for all other communications types.

More information on the APPC communications type can be found in the *APPC Programmer's Guide*.

***SYS:** The advanced program-to-program communications (APPC) mapped conversation support is used.

***USER:** The advanced program-to-program communications (APPC) basic conversation support is used.

***SRCPGM:** The target program accepts the conversation type specified by the source program. If this value is specified, RMTLOCNAME(*REQUESTER) must also be specified.

BLOCK

Specifies that either the system or the user controls whether records are combined into blocks when they are sent. This parameter applies to the BSCEL communications type only and is ignored for all other communications types.

With this parameter, the user may specify one of the following conditions of record formatting:

- No blocking or deblocking: The record format described in the DDS is the format for both the record and the block.
- User blocking and/or deblocking: Gives the BSC controls needed to describe the record format of the system.
- System blocking with record separator characters: Specify the record separator character used by the system to determine record boundaries within the block.
- System blocking of fixed-length records: The system uses fixed-length records, and blocks and/or deblocks records accordingly.

If a parameter value other than *NONE or *USER is specified, records are blocked for output and are deblocked upon input as required by the system.

Element 1: Blocking Options

***DEVD:** The block option specified in the device description is used.

***NONE:** Blocking or deblocking is not done by the system.

***ITB:** The records are blocked or deblocked, based on the location of an intermediate text block (ITB) control character. For input files, a record is delimited by locating the next intermediate text block character. An end-of-text or end-of-transmission block character is used as an intermediate text block character to delimit a block. For output files, an ITB character is added after the record. If it is the last character of the block, the ITB is replaced by an end-of-text or end-of-transmission block character.

***IRS:** The records are blocked or deblocked based on the location of an interrecord separator (IRS) character. For input files, a record is delimited by locating the next IRS character. For output files, an IRS character is added after the record.

***NOSEP:** No record separator character is contained in the block that is sent to or received from the device. The system blocks and deblocks the records by using a fixed-length record, as specified in the DDS format specifications.

***USER:** The program provides all the control characters (including record separator characters, binary synchronous communications (BSC) framing characters, and transparency characters) necessary to send records. More information about the device and binary synchronous communications equivalence link (BSCEL) support characteristics is in the *BSC Equivalence Link Programmer's Guide*.

***SEP:** The records are blocked or deblocked based on the location of a record separator character specified by the user. For input files, a record is delimited by locating the next record separator character. For output files, a record separator character is added after the record.

Element 2: Record Separator

X'1E': The record separator character X'1E' is used.

record-separator-character: Specify a record separator character that is unique and 1 byte in length. The record separator character may be specified as 2 hexadecimal characters, as in BLOCK(*SEP X'FD'), or the character may be specified as a single character by specifying a value ranging from 0 to 9 or A to F, as in BLOCK(*SEP A).

The following are dedicated BSC control characters that should not be used as record separator characters:

EBCDIC	ASCII	BSC Control
X'01'	X'01'	SOH (start-of-header)
X'02'	X'02'	STX (start-of-text)
X'03'	X'03'	ETX (end-of-text)
X'10'	X'10'	DLE (data-link escape)
X'1D'	X'1D'	IGS (interchange group separator)
X'1F'	X'1F'	ITB (intermediate text block)
X'26'	X'17'	ETB (end-of-transmission block)
X'2D'	X'05'	ENQ (enquiry)
X'32'	X'16'	SYN (synchronization)
X'37'	X'04'	EOT (end-of-transmission)
X'3D'	X'15'	NAK (negative acknowledgment)

RCDLEN

Specifies the maximum record length (in bytes) for data sent and received. This parameter applies to the BSCEL and the SNUF communications types only and is ignored for all other communications types.

***DEVD:** The record length specified in the device description is used. If a record is longer than the specified record length, a run time error occurs at the time the record is sent or received.

record-length: Specify the maximum record length (in bytes) to use with this device file. The value must be at least the size of the largest record sent. If a record is longer than the specified record length, a run time error occurs when the record is sent or received. Valid values range from 1 through 32767 bytes for SNUF communications. For BSCEL communications, the maximum record length is 8192 bytes.

For BSCEL communications, the maximum record length is 8192.

BLKLEN

Specifies the maximum block length (in bytes) for data sent. This parameter applies to the BSCEL and the SNUF communications types and is ignored for all other communications types.

***DEVD:** The block length specified in the device description is used.

block-length: Specify the maximum block length (in bytes) of records sent. The value must be at least the size of the largest record sent. Valid values range from 1 through 32767 for SNA upline facility (SNUF). For binary synchronous communications equivalence link (BSCEL) communications, the maximum block length is 8192.

TRNSPY

Specifies whether data is sent in transparent text mode. Text transparency allows all 256 extended binary-coded decimal interchange code (EBCDIC) character codes to be sent; use this function when sending packed or binary data fields. This parameter applies to the BSCEL communications type only and is ignored for all other communications types.

***DEVD:** The text transparency option specified in the device description is used.

***NO:** Text transparency is not used.

***YES:** Text transparency is used, which allows all 256 EBCDIC character codes to be sent. *YES is valid only if BLOCK(*NONE), BLOCK(*NOSEP), or BLOCK(*USER) is specified.

Note: Transparency of received data is determined by the data stream; therefore, this parameter is not relevant for received data. If TRNSPY(*YES) is specified with BLOCK(*USER), BSCEL ignores the transparency indicator during write operations. Correct controls must be given with the data to get transparent sending of data. For example, the data-link escape (DLE) and start-of-text (STX) control characters must first be specified; the system provides the remaining control characters required for transparent sending of data.

DTACPR

Specifies whether data compression is performed.

Note: This parameter applies to the BSCEL communications type only and is ignored for all other communications types.

***DEVD:** The data compression option specified in the device description is used.

***NO:** No data compression or decompression occurs. DTACPR(*YES) cannot be specified if TRNSPY(*YES) is specified.

***YES:** Data is compressed for output and decompressed for input.

TRUNC

Specifies whether trailing blanks are removed from output records. This parameter is for the BSCEL communications type only and is ignored for all other communications types.

***DEVD:** The truncation option specified in the device description is used.

***NO:** Trailing blanks are not removed from output records.

***YES:** Trailing blanks are removed from output records. TRUNC(*YES) cannot be specified if BLOCK(*NOSEP) or BLOCK(*ITB) is specified. If TRUNC(*YES) is speci-

fied and DTACPR(*YES) or BLOCK(*USER) is specified, truncation is ignored.

OVRFLWDTA
Specifies whether overflow data is discarded or retained.

***DISCARD:** Overflow data is not kept.

***RETAIN:** Overflow data is kept.

GRPSEP
Specifies a separator for groups of data (for example, data sets and documents). This parameter applies to the BSCEL communications types only and is ignored for all other communications types.

***DEVD:** The group separator option specified in the device description is used.

***DEV3740:** A null record (STXETX) is used as a data group separator.

***EOT:** A BSC control character EOT (end-of-transmission) is used as a data group separator.

***OFCSYS:** A sent block that ends with the BSC control character ETX (end-of-text) is used as a data group separator.

RMTBSCEL
Specifies the type of BSCEL session with the remote system. This parameter applies to the BSCEL communications types only and is ignored for all other communications types.

***DEVD:** The RMTBSCEL option specified in the device description is used.

***NO:** The remote system cannot recognize BSCEL commands or messages. In most cases, *NO is used when communicating with remote systems such as a 3741 Data Entry Station, an Office System 6, a 5230 Data Collection System, or a System/38.

***YES:** The remote system can recognize the BSCEL transaction starting commands, transaction ending commands, and online messages. In most cases, *YES indicates that the remote system is another AS/400 system, a System/38, a System/36, or a System/34 with BSCEL support.

INLCNN
Specifies the method used to make a connection on the line for the session being acquired. This parameter applies to the binary synchronous communications equivalence link (BSCEL) communications types only.

***CTLD:** The initial connection option specified in the controller description is used.

***ANS:** The remote system starts the call and the local system answers the call.

***DIAL:** The local system starts the call.

Examples

Example 1: Using RECID Keywords for Record Selection

```
ADDICFDEVE  FILE(ICFFILE1)
  PGMDEV(BSCEL2)  RMTLOCNAME(BSCNYC)
  FMTSLT(*RECID)
```

This command adds the program device entry named BSCEL2 with a corresponding remote location named BSCNYC for the ICF file ICFFILE1. The program device is added with the attributes of FMTSLT(*RECID).

Example 2: Using Remote Format Names for Record Selection

```
ADDICFDEVE  FILE(QGPL/ICFTEST)
  PGMDEV(APPC1)
  RMTLOCNAME(*REQUESTER)
  FMTSLT(*RMTFMT)  CNVTYPE(*SYS)
```

This command adds the program device entry named APPC1 with a remote location name of *REQUESTER for the ICF file ICFTEST in the QGPL library. This program device entry has the FMTSLT(*RMTFMT) and CNVTYPE(*SYS) attributes.

Example 3: Adding a Program Device Entry

```
ADDICFDEVE  FILE(ICFLIB/TESTFILE)
  PGMDEV(JOE)  RMTLOCNAME(LU0MPLS)
```

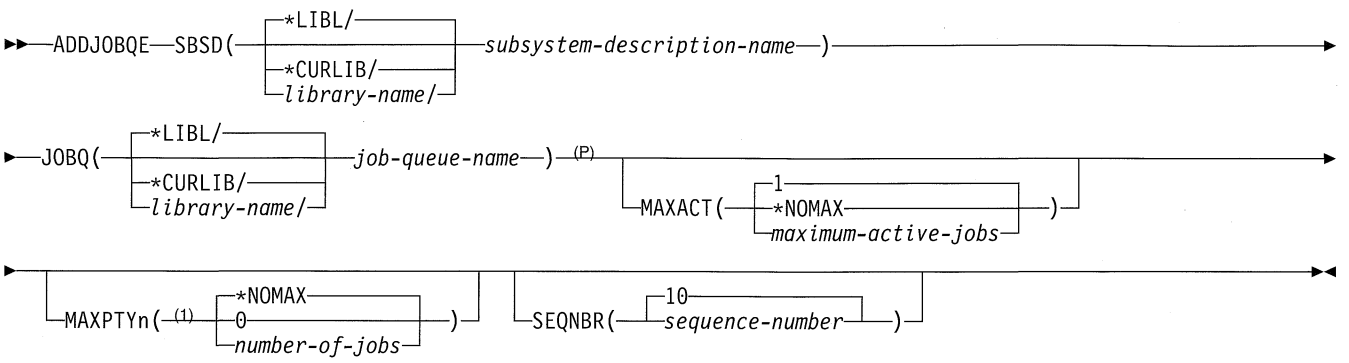
This command adds the program device entry named JOE with remote location named LU0MPLS for the ICF file TESTFILE in library ICFLIB.

Example 4: Adding a Program Device Entry

```
ADDICFDEVE  FILE(TESTFILE)
  PGMDEV(APPC)  RMTLOCNAME(APPCMPLS)
  DEV(MPLSLINE2)
```

This command adds the program device entry named APPC with a remote location name of APPCMPLS using device MPLSLINE2 to the ICF file TESTFILE.

ADDJOBQE (Add Job Queue Entry) Command



Notes:

P All parameters preceding this point can be specified in positional form.

1 n equals 1 to 9.

Purpose

The Add Job Queue Entry (ADDJOBQE) command adds a job queue entry to the specified subsystem description. The associated subsystem must be inactive at the time. A job queue entry identifies the job queue from which jobs are selected for running in the subsystem. Jobs can be placed on a job queue by spooling readers or by using the following commands:

- Submit Database Jobs (SBMDBJOB)
- Submit Diskette Jobs (SBMDKTJOB)
- Submit Job (SBMJOB)
- Transfer Job (TFRJOB)
- Transfer Batch Job (TFRBCHJOB)

In a subsystem, job queues with lower sequence numbers are processed first. For more information, refer to the description of the SEQNBR parameter.

Restrictions: To use this command, the user must have object operational and object management authorities for the specified job queue. The specified job queue must already exist in the system if the library qualifier is not given. A job queue is created by the Create Job Queue (CRTJOBQ) command.

Required Parameters

SBSD

Specifies the qualified name of the subsystem description to which the job queue entry is added.

The name of the subsystem description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

subsystem-description-name: Specify the name of the subsystem description.

JOBQ

Specifies the qualified name of the job queue that is a source of batch jobs that are started by the subsystem. If the job queue does not exist when the entry is added, a library qualifier must be specified because the job queue name is retained in the subsystem description.

The name of the job queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

job-queue-name: Specify the name of the job queue.

Optional Parameters

MAXACT

Specifies the maximum number of jobs that can be active at the same time from this job queue. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

1: Only one job from the job queue can be active at any one time. However, the maximum activity level of the

routing entries might prevent routing steps from being started. If *NOMAX is specified, all the jobs on the job queue are started (within the limit specified by the MAXJOBS parameter in the subsystem description), even though the activity level of the storage pool being used might prohibit them from running at the same time.

***NOMAX:** There is no maximum number of jobs that can be active at the same time.

maximum-active-jobs: Specify the maximum number of jobs from this job queue that can be active at the same time.

MAXPTYn

Specifies the number of jobs that can be started for a specified job priority level.

***NOMAX:** There is no disconnect limit.

0: No jobs are started from a specified priority level.

number-of-jobs: Specify the number of jobs started in a specified priority level. Valid values range from 1 through 99.

SEQNBR

Specifies a job queue sequence number, which is used by the subsystem to determine the order in which the job queues are processed.

10: A sequence number of 10 is assigned to this job queue.

sequence-number: Specify the sequence number assigned to the job queue. The sequence number must be unique in the subsystem description. Valid values range from 1 through 9999.

The subsystem first selects jobs from the job queue with the lowest sequence number. When all jobs on that queue have been processed or the number of jobs specified on the MAXACT parameter has been reached, the subsystem processes jobs on the queue with the next higher sequence number. This process continues until all job queue entries have been processed or until the subsystem has reached its limit for overall maximum jobs (as specified on the MAXJOBS parameter in the subsystem description). In some cases, the sequence is interrupted and the subsystem processes a queue with a lower sequence number. This occurs for this subsystem when one of the following conditions occurs:

- A held job or job queue is released.
- A job is placed on or transferred to a queue.
- A new queue is allocated.
- A job ends.

Examples

Example 1: Adding a Job Queue

```
ADDJOBQE  SBSD(QGPL/NIGHTSBS)  JOBQ(QGPL/NIGHT)
  MAXACT(3)
```

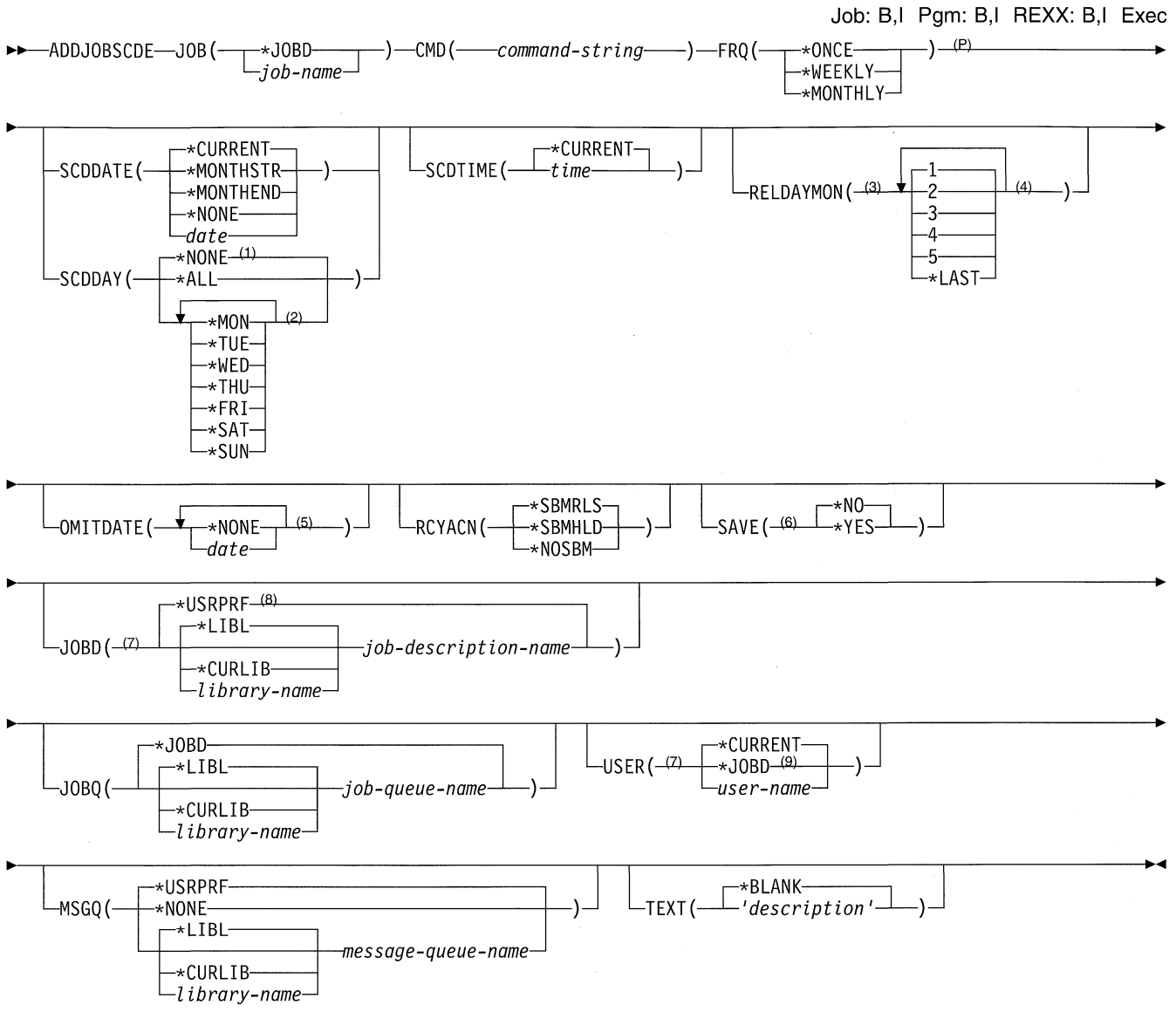
This command adds a job queue entry for the NIGHT job queue (in the QGPL library) to the NIGHTSBS subsystem description contained in the QGPL library. The entry specifies that up to three batch jobs from the NIGHT job queue can be active at the same time in the subsystem. The default sequence number of 10 is assumed.

Example 2: Running Jobs in Specific Priority Levels

```
ADDJOBQE  SBSD(QBASE)  JOBQ(JOBQ1)  MAXPTY1(2)
  MAXPTY7(0)  MAXPTY8(0)  MAXPTY9(0)
```

This command controls the selection of jobs for the job queue named JOBQ1 by setting priority levels to the values specified. This prevents any jobs with priority levels 7 through 9 from running.

ADDJOBSCDE (Add Job Schedule Entry) Command



Notes:

P All parameters preceding this point can be specified in positional form.

1 SCDDAY(*NONE) is not valid when SCDDATE(*NONE) is specified.

2 A maximum of 7 repetitions

3 Valid only if SCDDAY and FRQ(*MONTHLY) are specified.

4 A maximum of 5 repetitions

5 A maximum of 20 repetitions

6 Valid only if FRQ(*ONCE) is specified.

7 JOBD(*USRPRF) and USER(*JOBD) are mutually exclusive.

8 Not valid if USER(*JOBD) is specified.

9 Not valid if JOBD(*USRPRF) is specified or if the job description specifies USER(*RQD).

Purpose

The Add Job Schedule Entry (ADDJOBSCDE) command allows you to schedule batch jobs by adding an entry to the job schedule. You can use this command to schedule a batch job to be submitted once, or to schedule a batch job to be submitted at regular intervals.

The job schedule entry contains all the information needed to submit the job, including the command that the job runs, the job description and user profile under which the job is run, the job queue to which the job is submitted, and the message queue to which messages are sent.

The job is submitted to the specified job queue at the date and time specified on this command. This does not guarantee, however, that the job begins running at the scheduled time. The job does not begin running if the job queue is held or attached to an inactive subsystem, or if the maximum number of active jobs allowed to run in the subsystem or on the system at one time has been reached.

Each job schedule entry is identified by the job name, which is specified on the JOB parameter of this command, and an entry number, which is assigned by the system when the entry is added. The message replacement text for the message sent when an entry is added contains the entry number. If there is more than one entry with the same job name, you may need to specify the number when changing the entry using the Change Job Schedule Entry (CHGJOBSCE) command, removing the entry using the Remove Job Schedule Entry (RMVJOBSCDE) command, or holding or releasing the entry using the Hold Job Schedule Entry (HLDJOBSCDE) or Release Job Schedule Entry (RLSJOBSCDE) command. You can use the Work with Job Schedule Entries (WRKJOBSCDE) command to show or print entries.

More information is in the *Work Management Guide*.

Restrictions: 1) The user must have *USE authority to the job description and the user profile. 2) The user must have *USE and *ADD authorities to the message queue 3) The user must have *READ authority to the job queue and to all libraries associated with the specified objects.

Required Parameters

JOB

Specifies the name of the job schedule entry.

***JOBD:** The job description specified on the JOBD parameter is used for the name of the job schedule entry.

job-name: Specify the name of the job schedule entry.

Note: To avoid deleting, holding, or releasing entries created by IBM products when you are using generic names to delete, hold, or release your entries, do not add entries with job names beginning with the letter Q.

CMD

Specifies the command that runs in the submitted job. The IBM-supplied default routing program QCMD must be used when the job is started or the job will not run. Because the command you specify is used for the request data, the value specified on the RQSDTA parameter in the job description is ignored. The command you specify is syntax-checked when the entry is added.

You can specify a maximum of 512 characters.

FRQ

Specifies how often the job is submitted to run.

***ONCE:** The job is submitted once.

***WEEKLY:** The job is submitted on the same day or days of each week at the scheduled time.

***MONTHLY:** The job is submitted on the same day or days of each month at the scheduled time.

If you specify *MONTHLY and a month does not contain the date specified on the SCDDATE parameter, the job is not run that month. For example, if SCDDATE(01/31/93) and FRQ(*MONTHLY) are specified, the job is submitted on 01/31, 03/31, 5/31, 7/31, 8/31, 10/31, and 12/31, but will not run in February, April, June, September, or November. To submit a job on the last day of every month, specify SCDDATE(*MONTHEND).

If you specify *MONTHLY and your system or your job is configured to use Julian date format, the job is submitted to run on the day of the month that it would run if the system or job did not use Julian date format.

Optional Parameters

SCDDATE

Specifies the date on which the job is submitted to run.

If your system or your job is configured to use the Julian date format, the *MONTHSTR and *MONTHEND values are calculated as if the system or job did not use the Julian date format.

***CURRENT:** The current date is used.

***MONTHSTR:** The job is submitted on the first day of the month. If you specify *MONTHSTR, and if today is the first day of the month, and if the time you specify on the SCDTIME parameter has not passed, the job is submitted today. Otherwise, the job is submitted on the first day of the next month.

***MONTHEND:** The job is submitted on the last day of the month. If you specify *MONTHEND, and if today is the last day of the month, and if the time you specify on the SCDTIME parameter has not passed, the job is submitted today. Otherwise, it is submitted on the last day of the next month.

***NONE:** No date is specified for the job to be submitted.

date: Specify the date in the job date format.

SCDDAY

Specifies the day of the week on which the job is submitted.

If today is the day of the week specified on this parameter and the time specified on the SCDTIME parameter has not passed, the job is submitted today. Otherwise, the job is submitted on the next occurrence of the specified day. For example, if SCDDAY(*FRI) and SCDTIME(12:00:00) are specified, and you are adding this job schedule entry at 11:00 a.m. on a Friday, the job is submitted today. If you are adding the entry at 4:00 p.m. on a Friday, or at 11 a.m. on a Monday, the job is submitted the following Friday.

***NONE:** No day is specified for the job to be submitted.

***ALL:** The job is submitted every day.

***MON:** The job is submitted on Monday.

***TUE:** The job is submitted on Tuesday.

***WED:** The job is submitted on Wednesday.

***THU:** The job is submitted on Thursday.

***FRI:** The job is submitted on Friday.

***SAT:** The job is submitted on Saturday.

***SUN:** The job is submitted on Sunday.

SCDTIME

Specifies the time on the scheduled date at which the job is submitted to run.

Note: Although the time can be specified to the second, the activity involved in submitting a job and the load on the system may affect the exact time at which the job is submitted.

***CURRENT:** The job is submitted at the current time. If you specify SCDTIME(*CURRENT) and SCDDATE(*CURRENT), the job is immediately submitted to the specified job queue.

time: Specify the time. The time is specified in 24-hour format with or without a time separator as follows:

- With a time separator, specify a string of 5 or 8 digits where the time separator separates the hours, minutes, and seconds. If this command is entered from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command fails.
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds. Valid values for **hh** range from 00 through 23. Valid values for **mm** and **ss** range from 00 through 59.

RELDAYMON

Specifies the relative day of the month on which the job is submitted to run.

You can specify a value on this parameter only if the SCDDAY parameter and FRQ(*MONTHLY) are specified.

1: The job is submitted on the specified day of the week the first time it occurs in the month. For example, if you specify SCDDAY(*TUE), FRQ(*MONTHLY), and RELDAYMON(1), the job is submitted on the first Tuesday of every month.

2: The job is submitted on the specified day the second time it occurs in the month.

3: The job is submitted on the specified day the third time it occurs in the month.

4: The job is submitted on the specified day the fourth time it occurs in the month.

5: The job is submitted on the specified day the fifth time it occurs in the month.

***LAST:** The job is submitted on the specified day the last time it occurs in the month.

OMITDATE

Specifies a list of dates on which the job is not submitted. You can, for example, use this parameter to prevent recurring jobs from running on holidays. The date must be specified in the job date format.

***NONE:** No dates are specified when a job is not submitted.

date: Specify a date when a job is not submitted.

RCYACN

Specifies the recovery action to be taken if the scheduled job cannot be submitted at the designated time because the system is powered down or in restricted state. The action specified on this parameter occurs at the next initial program load (IPL) or when the system comes out of restricted state.

Jobs submitted during IPL or when the system comes out of restricted state are submitted in the order that they would have been had they been submitted at the times specified in the job schedule entries. If multiple occurrences of a recurring job are missed, the job is submitted only once. The first missed occurrence of a recurring job is used to order the jobs. The next occurrence of the job is calculated from the current date.

Since the scheduler portion of IPL need not be complete for the IPL of the system to be complete, other jobs may start on the system before all of the scheduled jobs have been submitted.

This parameter does not apply in the following instances:

- When a job is released after being held at the date and time it was to be submitted
- When the date and time at which a job is to be submitted passes because of changes to date and time system values

***SBMRLS:** The job is submitted in the released (RLS) state.

***SBMHLD:** The job is submitted in the held (HLD) state.

***NOSBM:** The job is not submitted.

Specifying *NOSBM affects only missed occurrences of the job. If the job schedule entry is a recurring job, future occurrences are not affected.

SAVE

Specifies whether the entry for a job that is submitted only once is kept after the job is submitted. This parameter is valid only if FRQ(*ONCE) is specified.

***NO:** The entry is not kept after the job is submitted.

***YES:** The entry is kept after the job is submitted. If you specify *YES, the job is submitted once. The job is not submitted again until the Change Job Schedule Entry (CHGJOBSCDE) command is used to specify a new date and time.

JOBD

Specifies the qualified name of the job description used when submitting the job.

***USRPRF:** The job description specified in the user profile under which the submitted job runs is used. The user profile is specified on the USER parameter.

The name of the job description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

job-description-name: Specify the name of the job description.

JOBQ

Specifies the qualified name of the job queue on which this job is placed.

You must have authority to the queue to specify a name on this parameter. Authority to the queue cannot be received through program adoption.

***JOBD:** The scheduled job is placed on the job queue specified in the job description. The job description is specified on the JOBD parameter.

The name of the job queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

job-queue-name: Specify the name of the job queue.

USER

Specifies the name of the user profile under which the scheduled job is submitted.

***CURRENT:** The user profile under which the current job is running is used.

***JOBD:** The user profile specified in the job description is used for the job schedule entry.

user-name: Specify the name of the user profile that is used. You must be authorized to the user profile. The user profile must be authorized to the job queue, job description, and message queue specified on this command.

MSGQ

Specifies the qualified name of the message queue to which messages are sent.

Messages are sent when the job is submitted and when a submitted job has completed running. Messages indicating a serious error are sent to the QSYSOPR message queue regardless of the value specified on this parameter when:

- The message queue specified on this parameter is damaged.
- MSGQ(*NONE) is specified.
- MSGQ(*USRPRF) and USER(*JOBD) are specified, and the job description specified on the JOBD parameter is changed to USER(*RQD) after the entry is added.

Note: When MSGQ(*USRPRF) is specified and the user profile contains a message queue name with *LIBL specified for the library, the results can be unpredictable. When the job is submitted, the library list from the system value object is used.

***USRPRF:** The message queue specified in the user profile under which the submitted job runs is used. The user profile is specified on the USER parameter.

***NONE:** Completion messages are not sent. Error messages are sent to the QSYSOPR message queue.

The name of the message queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

message-queue-name: Specify the name of the message queue to which messages are sent.

TEXT

Specifies text that briefly describes the job schedule entry. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Examples

Example 1: Scheduling a Weekly Job

```
ADDJOBSCDE  JOB(CLEANUP)  SCDDATE(*NONE)
  CMD(CALL  PGM(CLNUPLIB/CLNUPPGM))
  SCDDAY(*FRI)  SCDTIME('23:00:00')
  FRQ(*WEEKLY)  RCYACN(*NOSBM)
  JOBD(CLNUPLIB/CLNUPJOBD)
```

This command submits a job named CLEANUP every Friday at 11 p.m. The job uses job description CLNUPJOBD in library CLNUPLIB. If the system is powered down or is in the restricted state at 11 p.m. on Friday, the job is not submitted at IPL or when the system comes out of restricted state.

Example 2: Scheduling a Monthly Job

```
ADDJOBSCDE   JOB(PAYROLLJOB)  CMD(CALL PAYROLL)
  SCDDATE(*NONE)  SCDDAY(*MON)
  SCDTIME('09:00:00')  FRQ(*MONTHLY)
  RELDAYMON(1)
```

This command submits a job to run program PAYROLL at 9 a.m. on the first Monday of every month.

Example 3: Omitting Dates

```
ADDJOBSCDE   JOB(MONTHEND)
  CMD(CALL INVENTORY)
  SCDDATE(*MONTHEND)  SCDTIME('23:30:00')
  FRQ(*MONTHLY)  OMITDATE('12/31/93')
```

This command submits a job to run program INVENTORY at 11:30 p.m. on the last day of every month except December 31, 1993.

Example 4: Scheduling a Daily Job

```
ADDJOBSCDE   JOB(*JOBD)  CMD(CALL DAILYCLEAN)
  SCDDATE(*NONE)  SCDDAY(*ALL)
  SCDTIME('18:00:00')  FRQ(*WEEKLY)
  RCYACN(*NOSBM)  USER(SOMEPGMER)
```

This command submits a job to run program DAILYCLEAN every day at 6 p.m. The job runs under user profile SOMEPGMER. If the system is powered down or is in the restricted state at 6 p.m., the job is not submitted at IPL or when the system comes out of restricted state.

Example 5: Scheduling a Weekly Job

```
ADDJOBSCDE   JOB(*JOBD)
  CMD(CALL PGM1)  SCDDATE('06/01/93')
  FRQ(*WEEKLY)  USER(PGMR1)
```

This command submits a job to run program PGM1 every week starting on June 1, 1993 at the current time. Because June 1 is a Saturday, the job is submitted every Saturday.

Example 6: Scheduling a Job to Run Twice a Month

```
ADDJOBSCDE   JOB(*JOBD)  CMD(CALL PGM2)
  SCDDATE(*NONE)  SCDDAY(*MON *WED)
  FRQ(*MONTHLY)  RELDAYMON(3)
  SCDTIME('23:30:00')
```

This command submits a job to run program PGM2 every third Monday and every third Wednesday at 11:30 p.m. The job is submitted this month if the third Monday and Wednesday have not passed when this entry is added. If, for example, yesterday was the third Monday, today is the third Tuesday, and tomorrow is the third Wednesday, the job is submitted tomorrow, and then not again until next month.

Example 7: Scheduling a Job to Run Twice a Month

```
ADDJOBSCDE  JOB(*JOBD)  CMD(CALL PGM3)
  SCDDATE(*NONE)  SCDDAY(*MON)
  FRQ(*MONTHLY)  RELDAYMON(1 3)
  SCDTIME('09:00:00')  USER(PGMR3)
```

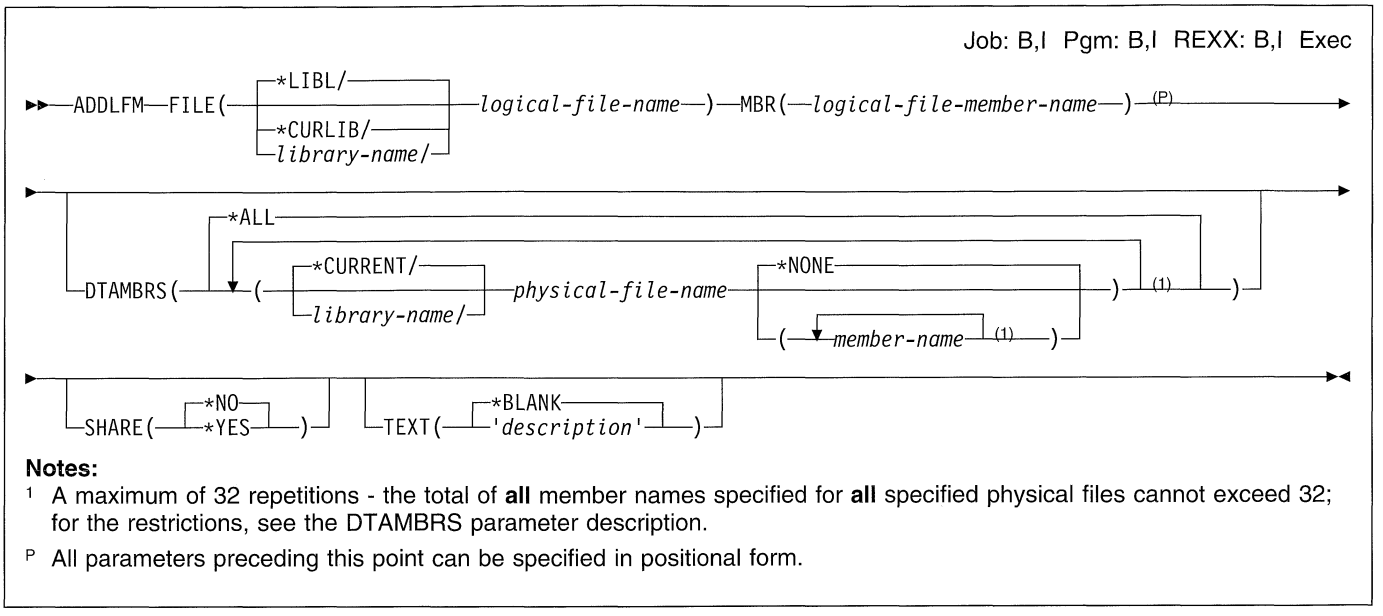
This command submits a job to run program PGM3 on the 1st and 3rd Monday of every month at 9:00 a.m. The job runs under user profile PGMR3.

Example 9: Scheduling a Job to Run Every Weekday

```
ADDJOBSCDE   JOB(*JOBD)  CMD('CALL PGM4')
  SCDDATE(*NONE)
  SCDDAY(*MON *TUE *WED *THU *FRI)
  SCDTIME('19:00:00')  FRQ(*WEEKLY)
```

This command submits a job to run program PGM4 every weekday at 7 p.m.

ADDLFM (Add Logical File Member) Command



Notes:

1 A maximum of 32 repetitions - the total of **all** member names specified for **all** specified physical files cannot exceed 32; for the restrictions, see the DTAMBRS parameter description.

P All parameters preceding this point can be specified in positional form.

Purpose

The Add Logical File Member (ADDLFM) command adds a named member to the specified logical file, which must already exist on the system. A member must be added to the logical file before the file can have access to data stored in any physical file member. The first member of a file can be added by entering an ADDLFM command or by specifying a member name in the MBR parameter of the Create Logical File (CRTLF) command. To add other members to the file, use the ADDLFM command to specify each one.

A logical file member can use the data from all, or a subset of, the physical files referenced by the logical file. Each member has its own set of data and can have its own access path that provides an organization to that data. The system attempts to implicitly share an access path already on the system.

The number of members that can be added to a logical file is limited to the number specified in the MAXMBRS parameter of the associated Create Logical File (CRTLF) command. Each added member has the same attributes as those defined in the logical file.

Restrictions: To add a member to a keyed logical file, the user must have object management authorities and object operational authorities for each of the physical files on which the logical file member is based (specified explicitly by the DTAMBRS parameter or implicitly by the PFILE or JFILE keyword specified in DDS). For a member added to a non-keyed logical file, object operational and object management authorities for each of the physical files is required.

Note: An *EXCLRD lock is required on the file to add a member. Because this command adds a member to a file in a library, the library must not be locked (*SHRNUP or *EXCLRD with the Allocate Object (ALCOBJ) command) in another job.

Required Parameters

FILE

Specifies the qualified name of the logical file to which this member is added.

The name of the logical file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

If a DDM file is specified, the target system specified must be an AS/400 system.

logical-file-name: Specify the name of the file to which the member is added.

MBR

Specifies the name of the logical file member that is being added. The member name must be unique in the file to which it is being added.

If the FILE parameter specifies a DDM file and a member name is specified as part of the remote file name in the DDM file, the member names must be the same.

Optional Parameters

DTAMBRS

Specifies the names of the physical files and members that contain the data associated with the logical file member being added by this command. A logical file member can be based on all of the physical files and members on which the logical file itself is based, specified by DTAMBRS(*ALL), or the member can be based on a subset of the total files and members, specified by DTAMBRS(qualified-file-names [member-names]).

For DDM file:

- The file names specified in the DTAMBRS parameter must be the names of the DDM files that represent the remote based-on physical files. If a member name was specified as part of the remote file name in the DDM file, only that member name can be specified on the DTAMBRS parameter. The member names must be the actual remote file member names.
- The based-on physical files must be at the same system location as the logical file to which the member is being added.
- If no member name is specified for the remote file name in the DDM file, all members are accessible. If only one member name is specified, only that member is accessible through that DDM file.

If the FILE parameter specifies a join logical file or an arrival sequence logical file, only one data member can be specified on the DTAMBRS parameter for each physical file that was specified on the PFILE or JFILE keyword in the DDS. *ALL is valid only if each physical file has only one member. If any of the physical files has more than one member, the specific physical file member must be specified on the DTAMBRS parameter.

The same physical file name can be specified more than once on the JFILE keyword. In this case, each occurrence of the file name is treated as a different based-on physical file, and must be specified on the DTAMBRS parameter.

If the logical file has more than one format and *ALL is not specified on the DTAMBRS parameter, the physical file or member must be specified on the DTAMBRS parameter the same number of times as it appears on the PFILE keyword of the DDS for the logical file. If the physical file or member is specified only once and multiple formats exist on the logical file, only the first format that uses the physical file or member in the PFILE keyword is used.

***ALL:** The logical file member being added is based on all the physical files and members (used by the logical file) that exist at the time this ADDLFM command is entered. At least one member must exist in at least one of the physical files. The physical file names are specified on the PFILE or JFILE keyword in the DDS.

physical-file-name member-name: Specify the names of the physical files and their members that contain the data to be accessed by the logical file member being added. The physical file names must match a name on the PFILE or JFILE keywords in the DDS for the logical file and cannot be specified more often on the DTAMBRS parameter than on the PFILE or JFILE keywords in the DDS. For join logical files, all physical files specified on the JFILE keyword must be specified on the DTAMBRS parameter and must contain one and only one member. If a physical file name is not specified for a file that is on a PFILE or JFILE keyword in the DDS, the logical file member is not based on any member of that physical file. Refer to "Additional Considerations" in the CRTLF command for more details.

Up to 32 qualified physical file names and physical file member names can be specified. Also, the total of *all* member names cannot exceed 32; that is, the member names specified for the files specified cannot be greater than 32. For example, one file can specify 32 members, two files can each have 16 members, or 32 files can each have one member specified.

If a logical file is created, the physical files specified on the PFILE or JFILE keyword are used to create the logical file. If no library name is specified for the physical files on the PFILE or JFILE keyword, the library list (*LIBL) at file creation time is used to find the physical files; the physical files from the library list are used to create the logical file. The qualified physical files from the PFILE or JFILE keyword (regardless of whether a library name was specified or the library list was used to find the files) are the physical files associated with the logical file. The name of the physical files associated with the logical file are saved in the description of the logical file. If a member is added to the logical file, the DTAMBRS parameter is used to specify the physical file members associated with the logical file member. Each physical file name specified on the DTAMBRS parameter must be associated with the logical file (saved in the description of the logical file). If a library name is not specified, the current library name (*CURRENT) from the logical file description is used. If the library name is specified, the physical file must be associated with the logical file. If the logical file is associated with more than one physical file of the same name, the library name must be specified.

The following examples show the syntax for specifying single and multiple members for single and multiple physical files. In the examples, the abbreviation PF represents a physical file name, LIB represents a library qualifier, and M represents a member name.

Single physical file and member:

```
DTAMBRS((PFA M1))
```

Single file with multiple members:

```
DTAMBRS(PFA (M1 M2 M3))
```

Multiple files with single members and no members:

```
DTAMBRS((PFA M1) (PFB M4) (PFE *NONE))
```

Multiple files with multiple members:

```
DTAMBRS((PFA (M1 M3 M4)) (PFB (M1 M2 M4)))
```
Multiple files with the same name in different libraries:
```
DTAMBRS((LIBX/PFA M1) (LIBY/PFA (M1 M2)))
```
Multiple files with the same name in the same library:
```
DTAMBRS((LIBX/PFA M1) (LIBX/PFA M1))
```

If more than one physical file member is specified for a physical file, the member names are specified in the order in which records are retrieved if duplicate key values occur across those members.

SHARE

Specifies whether the open data path (ODP) for the logical file member is shared with other programs in the routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

More information on shared database files is in the *Database Guide*.

***NO:** The ODP created by the program with this attribute is not shared with other programs in the routing step. Every time a program opens the file with this attribute, a new ODP to the file is created and activated.

***YES:** The ODP created with this attribute is shared with each program in the routing step that also specifies SHARE(*YES) when it opens the file.

Note: When SHARE(*YES) is specified and control is passed to a program, a read operation in that program retrieves the next input record. A write operation produces the next output record.

TEXT

Specifies text that briefly describes the logical file member. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***BLANK:** Text is not specified.

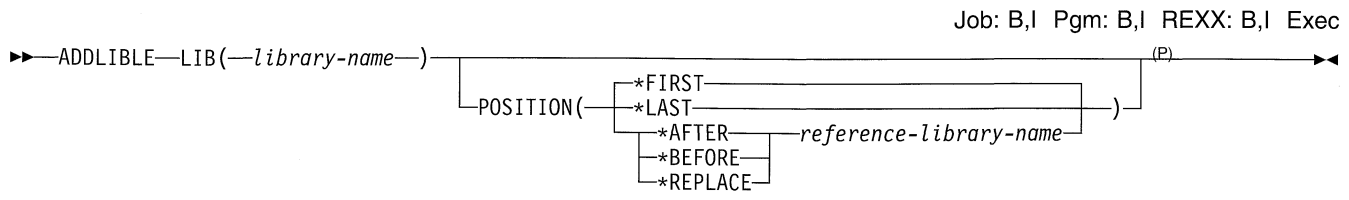
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
ADDLFM   FILE(INVENLIB/STOCKTXS)
  MBR(JANUARY)
  DTAMBRS((INVENTXS JANUARY))
  TEXT('JANUARY STOCK ACTIVITY BY LOCATION')
```

This command adds a member named JANUARY to the logical file named STOCKTXS in the INVENLIB library. The logical file has access to the data stored in the JANUARY member of the INVENTXS physical file.

ADDLIBLE (Add Library List Entry) Command



Note:

| P All parameters preceding this point can be specified in positional form.

Purpose

The Add Library List Entry (ADDLIBLE) command adds a library name to the user portion of the library list (after the current library list entry if it exists) for the process in which the command was entered. The user can specify whether the library is added to the beginning or the end of the library list. In addition to this, the user can specify whether the library is added before, after, or replaces an existing library in the library list.

Required Parameter

LIB

Specifies the name of the library added to the user portion of the library list. Up to 25 libraries may exist in the user portion of the library list. Only one library name is added at a time with this command.

Optional Parameter

POSITION

Specifies the position in the user portion of the library list where the library is added.

***FIRST:** The library is inserted before the existing libraries in the user portion of the library list, after the current library, if it exists.

***LAST:** The library is added to the end of the user portion of the library list.

Element 1: Library Position

***AFTER:** The library specified on the LIB parameter is added to the user portion of the library list after the library specified on the POSITION parameter.

***BEFORE:** The library specified on the LIB parameter is added to the user portion of the library list before the library specified on the POSITION parameter.

***REPLACE:** The system clears the existing member and adds the new records.

Element 2: Reference Library Name

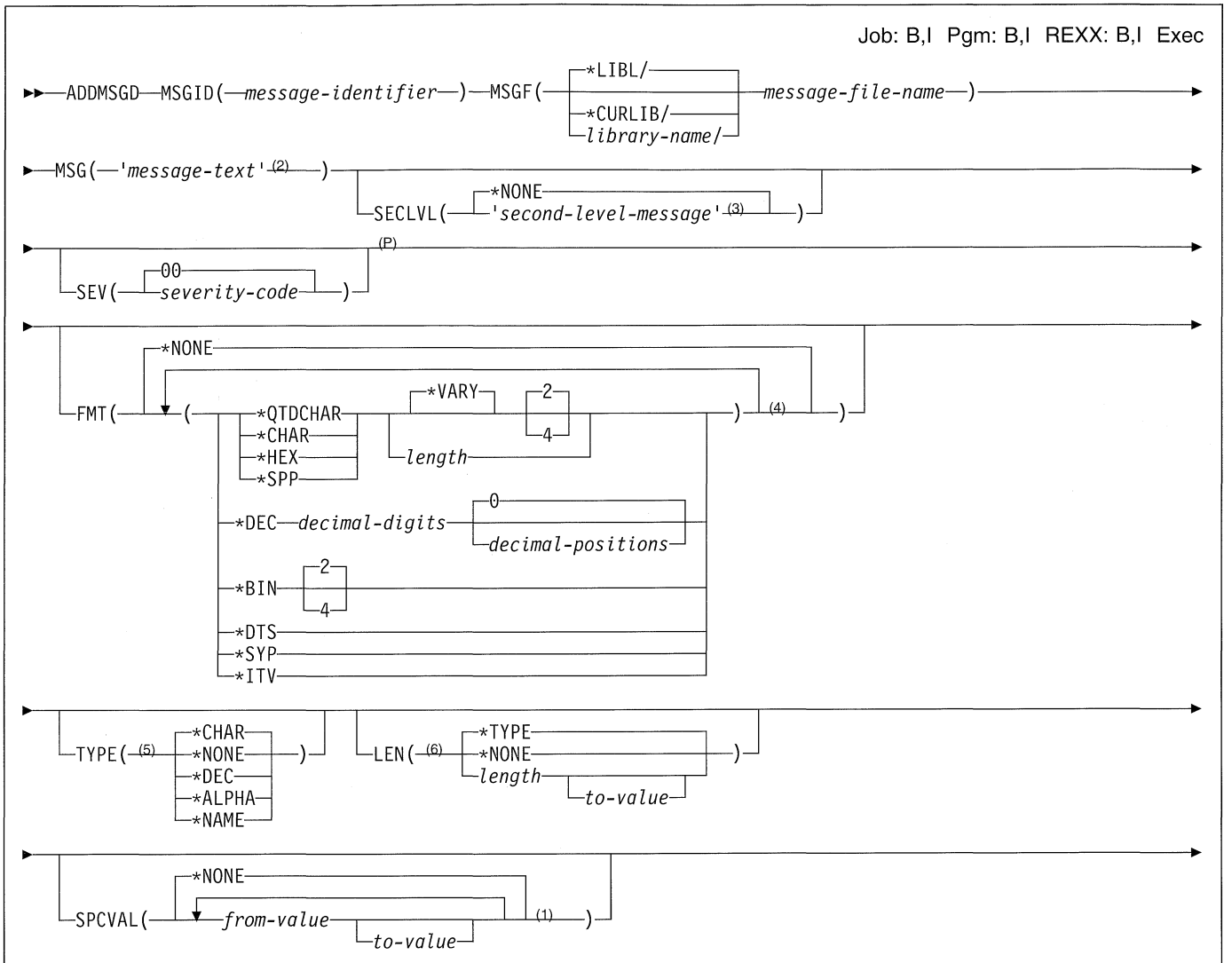
reference-library-name: Specify the name of a library that already exists in the user portion of the library list. The library specified in the LIB parameter is added after this library, before this library, or replaces this library in the user portion of the library list.

Example

```
ADDLIBLE  LIB(TESTLIB)  POSITION(*LAST)
```

This command adds the library TESTLIB to the end of the user portion of the library list.

ADDMSGD (Add Message Description) Command



Notes:

1 A maximum of 20 repetitions

2 No more than 132 characters can be specified.

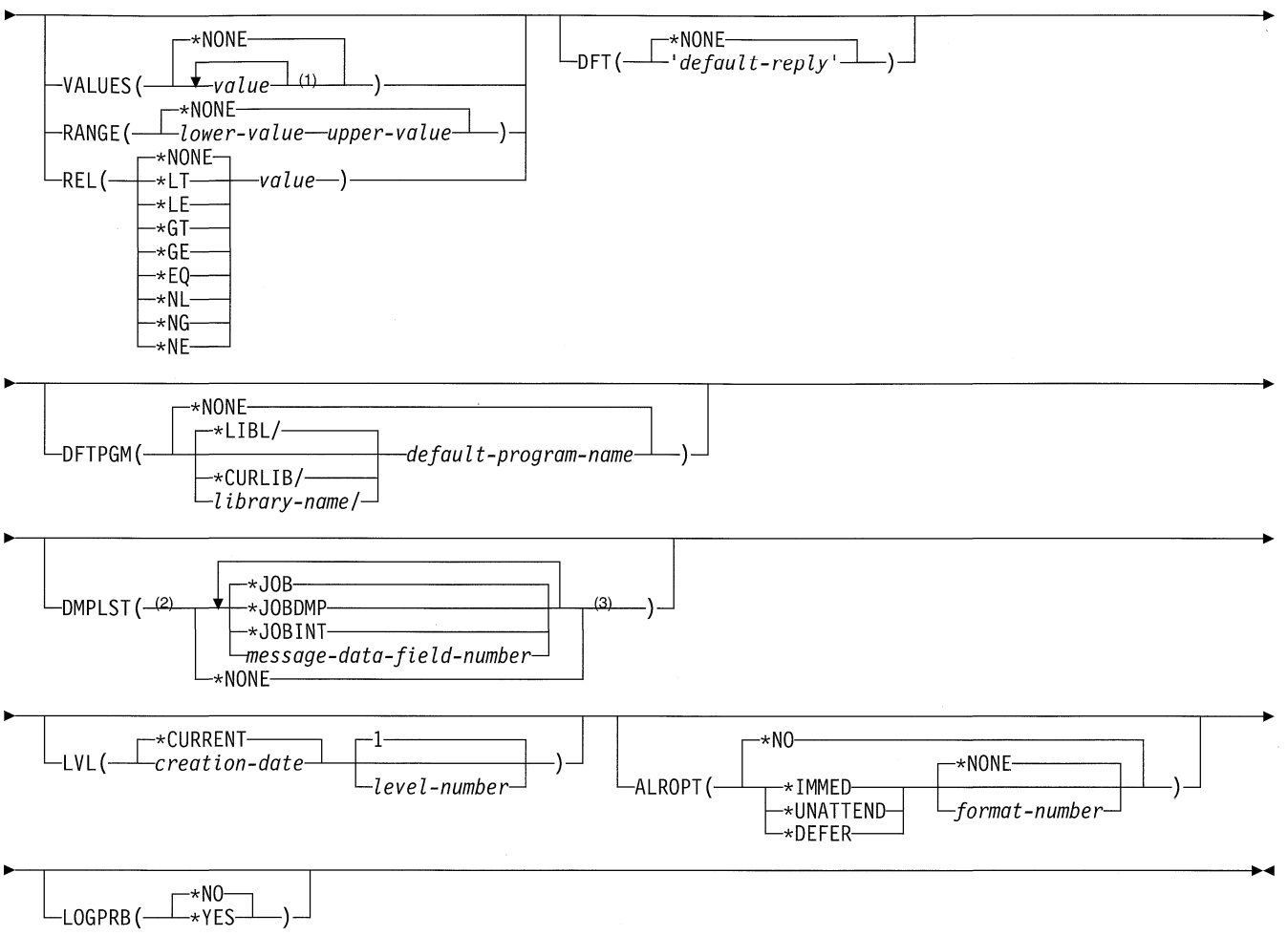
3 No more that 3000 characters can be specified.

P All parameters preceding this point can be specified in positional form.

4 A maximum of 99 repetitions

5 If TYPE is specified as *NONE, LEN must be *NONE also.

6 If LEN is specified as *NONE, TYPE must be *NONE also.



Notes:

1 A maximum of 20 repetitions

2 If any of the parameter values are specified for DMPLST, *JOB is assumed to be part of the value(s).

3 A maximum of 102 repetitions

Purpose

The Add Message Description (ADDMSGD) command describes a message and stores it in a message file for later use. The message description remains in the message file until the file is deleted or until the Remove Message Description (RMVMSGD) command is used to remove it from the file. To change any of the attributes of the message description, such as its message text or severity code, use the Change Message Description (CHGMSGD) command.

Note: A description of how to print a single message or a group of messages is in the section entitled "Handling Messages" in the *Operator's Guide*.

Substitution variables can be embedded both in the first-level and second-level message text. They can be replaced later by message data fields specified in the Retrieve Message (RTVMSG), Send User Message (SNDUSRMSG), and Send Program Message (SNDPGMMSG) commands.

Note: The *type* of message being defined is *not* specified in the ADDMSGD command. The type is specified in the command that actually sends the message.

If the message and its second-level text exceeds 512 characters, it will not fit in the prompt field. In this case, enter the command on the Command Entry panel or in a CL program.

Restriction: To add a message description to a message file, the user must have *USE and *ADD authorities to the message file.

Required Parameters

MSGID

Specifies the message identifier under which the message is stored in the message file. Every message must have a unique identifier.

The message identifier must be 7 characters long and made up of a message prefix and a message number, as shown in the following format:

```
pppnnnn
```

The first 3 characters (ppp) must consist of an alphabetic character followed by two alphanumeric (alphabetic or decimal) characters; the last 4 characters (nnnn) must consist of hexadecimal numbers ranging from 0 through 9 and A through F.

For **user-defined messages**, the same format *must* be used; in addition, the 3-character prefix should start with a U to distinguish user-defined messages from IBM-supplied messages. For example, the message identifier of a message in a payroll application message file could be UPY0027.

Following is a representative sample of the message identifier prefixes that are used to identify messages in several of the IBM-supplied message files. Shown are some of the prefixes for the OS/400 operating system, COBOL/400*, RPG/400*, the IDU and SDA utilities, and keyboard and machine interface (MI) messages.

CPA	OS/400 system operator action
CPC	OS/400 system completion messages
CPD	OS/400 system diagnostic messages
CPI	OS/400 system informational messages
CPX	OS/400 system titles and texts
CPZ	OS/400 system abnormal termination
CBE	COBOL run time
CBL	COBOL compiler
CBX	COBOL titles and texts
CSC	COBOL syntax checker
QRG	RPG language compiler
RPG	RPG run time
RPT	RPG auto report
RSC	RPG syntax checker
RTX	RPG auto report titles and texts
RXT	RPG relational diagnostic texts
IDU	Interactive database utilities (IDU)
IDX	IDU titles and text
SDA	Screen design aid (SDA)
SDX	Screen design aid titles and texts
KBD	Keyboard
MCH	AS/400 system machine instruction interface

MSGF

Specifies the qualified name of the message file where the message is to be stored. The IBM-supplied message files (for example, QCPFMSG and QRPGMSG) cannot be specified unless the user entering the command has authority to update those files. When the system is installed, only the system security officer has that authority. Message file overrides in effect for the job are ignored by this command and the message is stored in the specified file.

The name of the message file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

message-file-name: Specify the name of the message file to use.

MSG

Specifies the first level of message text of the message being defined. This text is the message that is initially shown or printed, or sent to a program or log. Up to 132 characters (enclosed in apostrophes) can be specified, but the limitations of the display stations (their screen size) should be considered. The entire message must be enclosed in apostrophes if blanks are included in the message. To code an apostrophe for use in the message, enter a double apostrophe.

One or more substitution variables can be embedded in the message text string to indicate positional replacement fields. These replacement fields allow substitution of variable data in the message by the program before the message is sent. The rules below must be followed if variables are used.

- Variables must be specified in the form &n, where n is a 1- or 2-digit (1-99) number identifying the data field that is substituted.
- Variables can be *preceded* by any alphanumeric character (including blanks). For example, the variables shown in the message below are valid.

  ```
  Command&34&72 &2 &99help
  ```

- Variables can be followed by any non-numeric character (the character following the variable cannot be digits 0-9). For example, the variables shown in the following example are *not* valid.

  ```
  Command&345 &244 &999help
  ```

- The variables can be enclosed in apostrophes if only the variables themselves make up the message. For example, to show a two-part decimal value, the message '&1.&2' can be specified.)
- Variables for this parameter do not have to be described positionally
  (in ascending or descending sequence).

Note: The data fields are described positionally in the FMT parameter and are specified positionally in the MSGDTA parameter of the Send Program Message (SNDPGMMSG) and Send User Message (SNDUSRMSG) commands. Details on substituting data fields in message text are in the *CL Programmer's Guide*.

'message-text': Specify the first-level message text, enclosed in apostrophes.

Double-Byte Character Set Considerations

If entering double-byte characters on this parameter, several combinations of characters may cause errors to occur on this command. If the double-byte characters contain the string, X'50Fn' (where n is a 1-digit number, ranging from 0 through 9), error messages CPF2424 or CPF2431 may be sent. Examples are: X'50F0', X'50F4', X'50F9'.

Optional Parameters

SECLVL

Specifies whether any second-level message text is shown to a display station user to further explain the message specified in the MSG parameter. The user presses the Help key to request the second-level message text. Second-level message text can also be written to the job log if *SECLVL is specified on the LOG parameter of the job commands.

***NONE:** There is no second-level message text for this message description.

'second-level-message': Specify the text shown as the second-level message text if it is requested by the user. No more than 3000 characters, enclosed in apostrophes, can be specified, but display limitations must be considered. One or more substitution variables can be embedded in the second-level message text, as described in the MSG parameter.

The second-level message text can be formatted for the display station using three format control characters:

Note: When formatting second-level message text for the display station, the format control characters must be followed by a blank space.

&N — Forces the text to a new line (column 2). If the text is longer than one line, the next lines are indented to column 4 until the end of the text or until another format control character is found.

&P — Forces the text to a new line, indented to column 6. If the text is longer than one line, the next lines start in column 4 until the end of the text or until another format control character is found.

&B — Forces the text to a new line, starting in column 4. If the text is longer than one line, the next lines are indented to column 6 until the end of the text or until another format control character is found.

Double-Byte Character Set Considerations

If entering double-byte characters on this parameter, several combinations of characters may cause errors to occur on this command. If the double-byte characters contain the string, X'50Fn' (where n is a 1-digit number, ranging from 0 to 9), error messages CPF2424 or CPF2431 may result. Examples are: X'50F0', X'50F4', X'50F9'.

SEV

Specifies the severity code of the message. The severity code indicates the severity level of the condition that causes the message to be sent.

00: The severity code assigned to this message is 00.

severity-code: Specify the severity level associated with this message. Valid values range from 00 through 99. The assigned code for the message should correspond in importance to the IBM-predefined severity codes. Any 2-digit value can be entered, even if no severity code (either predefined or user-defined) has been defined for it. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

FMT

Specifies the formats of up to 99 message data fields. Each field is described in this parameter by a list of attributes. All 99 of the message data fields can be used as substitution values in the first- and second-level message text defined in this message description. They also can be specified on the DMPLST or ALROPT parameter of this command. If specified in the MSGDTA parameter of the SNDPGMMSG or SNDUSRMSG commands, the data fields must be concatenated in one character string and must match the format and sequence specified. The length of the entire character string of concatenated message data fields cannot exceed 512 characters.

***NONE:** No format is being described for message fields. If *NONE is specified, or if this parameter is omitted, no references can be made to message data fields in the MSG, SECLVL, DMPLST, or ALROPT parameters.

type [length [decimal-positions]]: The format of each message data field (up to 99 fields) to be substituted in the message in this message description is defined by a list of attributes. These attributes specify the type of data in the field, the length of the field, and, optionally, the number of decimal digits to the right of the decimal point. Certain data types do not require a length field. Boundary alignment requirements must be considered (for example, pointers are always aligned on 16-byte boundaries).

Type of Message Data: The first value, type, specifies the type of data the substitution field contains and how the data is formatted when substituted in the message text. The contents of the second and third values vary depending on the type specified. One of the following types can be specified for each field described by this parameter:

***QTDCHAR:** A character string formatted with enclosing apostrophes ('Monday, the 1st') is used.

***CHAR:** A character string formatted without enclosing apostrophes is used. An alphanumeric string is used,

for example, to specify a name (BOB). Trailing blanks are truncated.

***HEX:** A string of bytes formatted as a hexadecimal value (for example, X'C0F4') is specified.

***SPP:** A 16-byte space pointer to data in a space object is specified. If referred to in the DMPLST parameter, the data in the space object (from the offset indicated by the pointer) for the length specified, is to be dumped. *SPP is not valid as a replacement field in message text.

***DEC:** A packed decimal number that is formatted in the message as a signed decimal value with a decimal point is used. Values for length (required) and decimal positions (optional) are specified for this type (*DEC) to indicate the number of decimal digits and the number of digits to the right of the decimal point. Zeros to the left of the first significant digit are suppressed, and leading blanks are truncated (removed). If a decimal position other than zero is specified, a decimal point is shown in the result even if the decimal precision in the result is zeros; examples are 128.00 and 128.01, if FMT(*DEC 5 2) is specified. If the number of decimal positions is not specified, zero is assumed. The following gives two examples:

- If FMT(*DEC 2) is specified for a substitution field and the message data is a packed decimal value of X'058C', the message text contains a positive value of 58 with no decimal point indicated.
- If FMT(*DEC 4 2) is specified and the packed value is specified as X'05810C' (3 bytes long), the text contains the formatted decimal value of 58.10.

***BIN:** A binary value that is either 2 or 4 bytes long (B'0000 0000 0011 1010') and is formatted in the message as a signed decimal value (58) is specified.

The following formats are valid only in IBM-provided message descriptions and should not be used for other messages.

***DTS:** An 8-byte field that contains a system date time stamp is specified. The date time stamp contains the date followed by one blank separator and the time. The output message date is formatted as specified by the job date format and job date separator. system values QDATFMT and QDATSEP. The time is formatted as hh:mm:ss.

***SYP:** A 16-byte system pointer to a system object is specified. If referred to in message text, the simple name of the system object is formatted as described in the name type, *CHAR. If referred to by the DMPLST parameter, the object itself is dumped.

***ITV:** An 8-byte binary field that contains the time interval (in seconds) for wait time-out conditions is specified. The time interval is formatted in the message as a zero-suppressed zoned decimal value (15 0) representing the number of seconds to wait.

Length of Message Data: Following the type specification, a second value (length) can be specified to indicate the number of characters or digits that are passed in the message data. How the second value is used depends on the type specified in the first value.

1. If a length is not specified for *QTDCHAR, *CHAR, *HEX, or *SPP, then *VARY is assumed for the length. If *VARY is specified or assumed, the message data field passed by the SNDUSRMSG or SNDPGMMSG commands must be preceded by a 2-byte or 4-byte binary field that indicates the actual number of bytes of data being passed. However, if *SPP is specified, the length field is contained in the first bytes pointed to by the space pointer. Therefore, the 2- or 4-byte field must precede the data pointed to by the space pointer, and must *not* precede the space pointer that is passed as part of the message data.
2. If the type *DEC is specified, the total number of decimal digits (including the fraction) *must* be specified as the second value; the number of digits in the fraction optionally can be specified as the third value.
3. If the type *BIN is specified, the message data field can be only 2 or 4 bytes long; the default value is 2 bytes.

Length Field Size/Decimal Positions: The third value is used in one of two ways, depending on the type specified in the first value. (1) If *QTDCHAR, *CHAR, *HEX, or *SPP is specified, and if *VARY is specified or assumed for the second value, the third value is used with *VARY to indicate the size of the length field actually passed. The third value can be either a 2 or a 4, which is the number of bytes used to specify the length (in binary) of the passed value. (2) If *DEC is specified, the third value indicates the number of decimal positions in the decimal value. If not specified for a decimal substitution value, the default is 0 decimal positions.

Note: If an object has been damaged or deleted, the substitution variable is not replaced by the object name if it is shown; instead, the variable appears as &n, where n = number. If the length of the message data that is passed to the substitution variable is shorter than the length specified for FMT, the substitution value becomes a null field.

If the message is an inquiry message (specified by *INQ in one of the send message commands) or a notify message (specified by *NOTIFY in the SNDPGMMSG command only) and a reply is expected, seven parameters can be used to specify some requirements that relate to the reply received. The seven validity checking parameters are: TYPE, LEN, VALUES, SPCVAL, RANGE, REL, and DFT.

These parameters are not necessary for a message to allow a reply, but they can be used to define valid replies that can be made to the message. Also note that the VALUES, RANGE, and REL parameters are mutually

exclusive; only one of them can be specified in this command.

TYPE

Specifies the type of reply that is valid to respond to an inquiry or notify message.

***CHAR:** Any character string is valid. If it is a quoted character string, the apostrophes are passed as part of the character string.

***NONE:** No reply type is specified, and no reply validity checking is performed. LEN(*NONE) must also be specified.

***DEC:** Only a decimal number is a valid reply.

***ALPHA:** Only an alphabetic (A through Z, $, #, and @) character string is valid. Blanks are not allowed.

***NAME:** Only a simple name is a valid reply. The name does not have to be an OS/400 system object name, but it must start with an alphabetic character; the remaining characters must be alphanumeric. If all reply characters are alphabetic (A-Z), the reply is converted to uppercase.

LEN

Specifies the maximum length of a reply to an inquiry or notify message. The values specified under *TYPE apply *only* if one or more of the other validity checking parameters are specified. If none of the validity checking parameters are specified, the reply type *CHAR can contain as many as 132 characters.

***TYPE:** The maximum length is determined by the type of reply specified on the TYPE parameter. The maximum length for each type of reply is as follows:

- Up to 132 characters can be specified for types *CHAR and *ALPHA. If additional validity checking is being performed (for example, if VALUES, RANGE, REL, or SPCVAL are specified), the maximum length allowed for *CHAR and *ALPHA is 32 characters.
- Up to 15 digits can be specified for *DEC, of which up to 9 digits can be to the right of the decimal point.
- Up to 10 alphanumeric characters can be specified for *NAME.

***NONE:** No reply length is specified. No reply validity checking is performed if this message is sent as an inquiry or notify message. TYPE(*NONE) must also be specified.

length [decimal-positions]: Specify the maximum length allowed for the message reply. The length specified cannot exceed the maximums specified on the *TYPE value. If the reply type is a decimal value, the number of decimal positions can be optionally specified; if it is not specified, zero decimal positions are assumed.

VALUES

Specifies a list of values of which one can be received as a valid reply to an inquiry or notify message. No more than 20 values can be specified in the list. Each value in the list must meet the requirements specified for message replies by the TYPE and LEN parameters.

If VALUES is specified, the RANGE and REL parameters cannot be specified. To be valid, a reply must match one of the values in this list.

For the reply value to match the compare value, both must be of the same keyboard shift. For example, if the program requires a reply containing uppercase characters, one of the following methods ensures a response in uppercase characters:

- Requiring a response in uppercase characters.
- Entering the compare values for the VALUES parameter in lowercase, but using the SPCVAL parameter to convert the characters to uppercase.
- Using the TYPE(*NAME) keyboard value to convert the characters to uppercase. To use this method, all reply characters must be alphabetic (A through Z).

***NONE:** No list of reply values is specified. The reply can have any value that is consistent with the other validity specification parameters.

value: Specify a list of up to 20 values to compare with a reply value that is sent in response to the message defined in this message description. The reply value must match one of the values in this list to be a valid reply to this message. The maximum length of each value is 32 characters.

SPCVAL

Specifies a list of up to 20 sets of special values of which one set (if the from-value is matched by the sent reply) is used as the reply for an inquiry or notify message. These values are special in that they may not meet all the validity checking specifications given in the other reply-oriented parameters. The reply sent is compared to the from-value in each set; if a match is found, and a to-value was specified in that set, the to-value is sent as the reply. If no to-value was specified, the from-value is sent as the reply. The to-value must meet the requirements specified in the TYPE and LEN parameters. If the reply sent does not match any from-value, the reply is checked for validity by the specifications in the other reply-oriented parameters.

***NONE:** No special values are specified for the replies to this message.

from-value [to-value]: Specify a list of up to 20 sets of special values to determine the reply to this message. Each set must have a from-value, with which the reply is compared, and an optional to-value, which is sent as the reply if its from-value matches the reply.

RANGE

Specifies the lower and upper value limits for valid replies to an inquiry or notify message. These values must meet the requirements specified for replies by the TYPE and LEN parameters, and both values must be of the same type. If both values are not of the same length, the shorter value is padded on the right with blanks. For the replies of types *CHAR and *ALPHA, the reply is padded on the right with blanks, or truncated on the right, to the length of the specified values, before the value range is checked for validity. If RANGE is specified, the VALUES and REL parameters cannot be specified.

***NONE:** No range values are specified for the replies to this message.

lower-value upper-value: Specify the upper and lower value limits for replies to an inquiry or notify message.

REL

Specifies the relation that must be met for a valid reply to an inquiry or notify message. The value specified must meet the requirements specified for replies by the TYPE and LEN parameters. For replies of the types *CHAR and *ALPHA, the reply is padded on the right with blanks, or truncated on the right, to match the length of the value specified, before the system performs the relational test on the reply value that is sent. If REL is specified, the VALUES and RANGE parameters cannot be specified.

***NONE:** No relational values are specified for the replies to this message.

operator-value: Specify one of the relational operators and the value against which the message reply is checked for validity. If the reply is valid in the relational test, it is sent to the message sender. The relational operators that can be entered are:

*LT	Less than
*LE	Less than or equal to
*GT	Greater than
*GE	Greater than or equal to
*EQ	Equal to
*NL	Not less than
*NG	Not greater than
*NE	Not equal to

DFT

Specifies, if the message is an inquiry or notify message, the default reply (enclosed in apostrophes, if it contains special characters) used if the receiver of the message has indicated that all incoming messages are to receive default replies, or if a message is deleted from a message queue and no reply was specified. The default reply can also be used to answer unmonitored notify messages. The default reply must meet the requirements specified for replies by the validity specification parameters, TYPE and LEN.

***NONE:** No default reply is specified for the replies to this message.

'default-reply': Specify the default reply to send (enclosed in apostrophes if it contains special characters) to inquiry or notify messages.

DFTPGM

Specifies the qualified name of any default program called to take default action if this message is sent as an escape message to a program that is not monitoring for it. This parameter is ignored if the message is *not* sent as an escape message. If it is sent as an escape message, the following parameters are passed to the specified default program:

- Program message queue name (10 characters). The name of the program message queue to which the message is sent. This name is the same as that of the program that did not monitor for the escape message.
- Message reference key (4 characters). The message reference key of the escape message on the program message queue.

***NONE:** No default program is specified for this message.

The name of the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

default-program-name: Specify the name of the default program that is called if an escape message is sent.

DMPLST

Specifies the data that is dumped if this message is sent as an escape or notify message to a program that is not monitoring for it. This parameter can specify that data related to the job be dumped, that data from message data fields be dumped, or that a combination of these be dumped. If data from message data fields is to be dumped, this parameter specifies one or more numbers that positionally identify the data fields to be dumped.

The system objects indicated by system pointers are completely dumped. The data in a space object, indicated by a space pointer, is dumped starting from the offset indicated by the space pointer for the length indicated in the field description. The standard job dump can also be requested. Dumps are taken as part of system default actions if escape messages are not monitored by the program to which they were sent.

***JOB:** This value has the same effect as specifying JOB(*) and OUTPUT(*PRINT) on the Display Job (DSPJOB) command. See the DSPJOB command description for more information.

***JOBDMP:** The data areas of the job are dumped as specified by the DMPJOB command. *JOBDMP can be specified by itself, with *JOB, *JOBINT, or a list of message data field numbers.

***JOBINT:** The internal machine data structures related to the machine process in which the job is running are dumped to the machine error log as specified by the DMPJOBINT command. *JOBINT can be specified by itself, with *JOBDMP, *JOB, or with a list of message data field numbers.

message-data-field-number: Specify the numbers of the message data fields that identify the data that is dumped if this escape message is sent but not monitored. Up to 99 data field numbers can be specified in the list; additionally, the list can contain the values *JOB, *JOBINT, and *JOBDMP.

***NONE:** There is no dump list for this message. No dump occurs.

Notes:

1. If any of these values are specified for DMPLST, *JOB is assumed to be part of the values. For example, DMPLST (1 2 *JOBDMP) gives the same result as DMPLST(*JOB 1 2 *JOBDMP).
2. Values specified for the DMPLST parameter may be overridden by the QSRVDMP system value. More information is in the chapter on system values in the *Work Management Guide*.
3. The program receiving the unmonitored message either must have a name starting with 'Q' or the message severity must be 50 or higher.
4. The user of the job in which the dump is specified must be authorized to the dump command requested on this parameter.

LVL

Specifies the level identifier of the message description being defined. The level identifier is made up of the date on which the message is defined and a 2-digit number that makes the identifier unique.

***CURRENT 1:** The current date and a 1 are used as the first and second parts of the message description level identifier.

creation-date level-number: Specify the date on which the message is being defined, and enter a 2-digit value (ranging from 1 through 99) that makes the level identifier of the message description unique. The date must be specified in the format defined by the system values QDATFMT and, if separators are used, QDATSEP.

ALROPT

Specifies the alert option associated with messages sent to the QHST log message queue. More information is in the *Communications and Systems Management Guide (Alerts and Distributed Systems Node Executive)*.

***NO:** No alert is sent.

Element 1: When an Alert is Sent

***IMMED:** An alert is sent immediately, simultaneous with sending the message to QHST.

***UNATTEND:** An alert is sent immediately only if the system is running in unattended mode.

The system is considered to be unattended if the Alert Status (ALRSTS) network attribute is set to *UNATTEND.

***DEFER:** The alert is sent after local problem analysis.

*DEFER should be specified only for messages against which a problem analysis can be run. An alert is sent at the first exit from problem analysis for the problem referred to by the message.

All alerts set to *DEFER are treated as *IMMED if:

- The system is running in unattended mode
- An error log ID is not available for a problem that might be resolved using problem analysis
- The LOGPRB parameter for the message is set to *NO

Element 2: Message Data Field Format Numbers

***NONE:** No message data field format number is passed with the alert identifier.

format-number: Specify the message data field format number that is passed with the alert identifier.

LOGPRB

Specifies, for IBM-supplied messages, whether an entry is put in the problem log. If there is an error log ID for the message and *YES is specified for this parameter, the user can call for problem analysis by pressing the F14 key from the system operator message queue display (Display Messages).

***NO:** An entry is not put in the problem log.

***YES:** An entry is put in the problem log if there is a error log ID associated with the message.

Examples

Example 1: Defining a Message

```
ADDMSGD   MSGID(UIN0115)  MSGF(INV)
  MSG('Enter the name of user''s department')
  SECLVL('Valid department names are:
  &B X12 &B X13 &B X14')
  TYPE(*CHAR)  LEN(3)  DFT('ZZZ')
```

This command defines a message and stores it in a file named INV under the identifier UIN0115. The message supplies second-level message text by using the &B formatting character to show the three valid department names (X12, X13, and X14) each on a separate line. The reply requires validity checking so that a valid reply can only be a 3-character identifier. A default reply of ZZZ is also provided.

Example 2: Defining a Message Description

```
ADDMSGD   MSGID(UPY0047)
  MSGF(PAYLIB/TIMECARD)
  MSG('For the week of &1, &2 time
  cards were processed.  Do you have more?')
  FMT((*CHAR 8)  (*CHAR 3))
  TYPE(*ALPHA)  LEN(1)  VALUES(N  Y)
  SPCVAL((YES Y)(NO N))  DFT(N)
```

This command defines a message description that is stored in the TIMECARD message file in the PAYLIB library. The program that processes the time cards can send a message (as an inquiry type message) telling how many time cards (in &2) have been processed for the week (specified in &1). To send this message to a user via a message queue, the program must use the SNDPGMMSG or SNDUSRMSG commands. In this example, the command specifies:

- The message identifier of this message (UPY0047)
- The file (TIMECARD) that contains this message
- The time card date in 8 characters (such as 09/15/88); this must be the first value in the MSGDTA parameter
- The number of time cards in no more than 3 digits (such as 125)

If a reply of YES is sent, it is accepted as a Y (SPCVAL parameter). If NO is sent, it is accepted as an N. If neither YES nor NO is sent, the reply is checked for validity by the TYPE, LEN, and VALUES parameters. If the user chooses, no reply is sent and the default reply (N) is assumed.

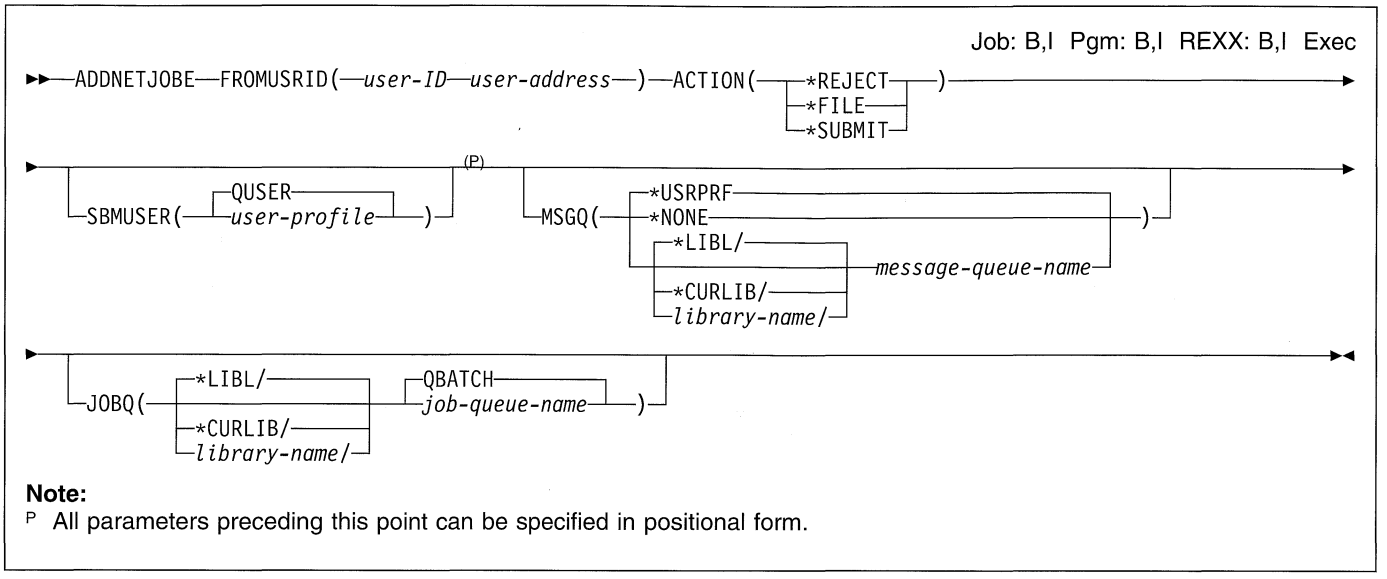
Example 3: Defining an Escape Message

```
ADDMSGD   MSGID(UPY1234)
  MSGF(PAYLIB/TIMECARD)
  MSG('Tax for employee &1 exceeds
  gross salary.')  SEV(75)
  FMT((*CHAR  6)(*DEC  9  2)(*CHAR  8))
  DFTPGM(PAYLIB/BADTAX)
  DMPLST(1  2  3 *JOB)
```

This command defines an escape message. The sender of the message passes three data values, the first of which (employee serial number) is used as replacement text in the message. If this message is sent as an escape message and the program to which the message is sent does not monitor for message UPY1234, default system action is taken. This includes dumping the three data values that were passed and the job structure. After the dump is taken, program BADTAX is called.

See the Monitor Message (MONMSG) command for more about monitoring for messages.

ADDNETJOBE (Add Network Job Entry) Command



Note:
P All parameters preceding this point can be specified in positional form.

Purpose

The Add Network Job Entry (ADDNETJOBE) command adds a network job entry to the network job table on the system. The network job entry is used to determine the action that is taken when an input stream is sent to a user on this system by using the Submit Network Job (SBMNETJOB) command. This entry determines whether the input stream is automatically submitted, placed on the queue of network files for a user, or rejected. The entry also specifies the user profile that is used for checking the authority to the job description referenced by the input stream. There must be one entry for each user or distribution group who intends to submit jobs to this system.

Note: There is a network attribute, JOBACN (Job Action), that provides overall control of network job submission. Its value must be *SEARCH before the network job table is searched for an action. If the network attribute is *REJECT, all incoming jobs are rejected. If the network attribute is *FILE, all incoming network jobs are saved in the user's queue of network files regardless of any network job entry. The network attribute can be changed with the Change Network Attributes (CHGNETA) command.

Each network job entry is identified by the two-part user ID of the sender. When an input stream arrives, the user ID of the sending user is used to find a network job entry. If no entry is found, the second part of the user ID is used to find an entry, using *ANY for the first part. If this search fails, a search is made using *ANY for both parts of the user ID. If no entry is found, the job is rejected.

For additional information on the network job table, refer to the *Distribution Services Network Guide*.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. The user must have *ALLOBJ (all object) authority.
3. The internal value for a node identifier may differ from the characters shown by the ADDNETJOBE command depending on the type of work station (language) being used. If the byte-string value specified for the FROMUSRID command parameter does not match the rules for an internal node identifier value, or if it does not match the internal value for any defined node (ignoring case differences), an error may be reported.

Required Parameters

FROMUSRID

Specifies the two-part user ID of the user who submits an input stream to this system. Any input streams received from the user are handled as specified in this network job entry. Both parts of the user ID are required. A special value of *ANY can be entered for the first part or for both parts of the user ID.

Note: Depending on the type of work station being used, the internal value for a user identifier may differ from the characters shown by the Display Network Job Entry (DSPNETJOBE) command. If the byte-string value specified for the FROMUSRID parameter does not match the rules for an internal user identifier value, or if it does not match the internal value for any enrolled user, an error may be reported.

ACTION

Specifies the action that is taken for the input stream controlled by this entry if the network attribute JOBACN is *SEARCH.

***REJECT:** The input stream is rejected.

***FILE:** The input stream is placed on the queue of network files for the user to whom the input stream is sent.

***SUBMIT:** The input stream is submitted to a batch job queue. The user profile specified in the network job entry is used to check for the required authority to the job queues that are used and to the job descriptions specified in the input stream.

Optional Parameters

SBMUSER

Specifies the user profile name under which jobs are submitted. This user profile name is used to check the authority to the job queues and job descriptions specified in the input stream. The value specified for this parameter will be effective if ACTION(*SUBMIT) is specified either on this command or on the Change Network Job Entry (CHGNETJOBE) command.

QUSER: The IBM-supplied user profile QUSER is used to submit the jobs.

user-profile: Specify the name of the user profile that is used to submit the jobs.

MSGQ

Specifies the qualified name of the message queue to which messages are sent.

Note: The message sent to the message queue specified here notifies the recipient that the input stream arrived and whether it was submitted, placed on the user's queue of network files, or rejected. A message is also sent to the history log (QHST) when an input stream arrives.

***USRPRF:** The message queue of the user profile to whom the job was sent is used. This user is specified on the TOUSRID parameter of the SBMNETJOB command; this may or may not be the same user as is specified on the SBMUSER parameter of this command.

***NONE:** No message is sent to a user; however, a message is sent to the history log (QHST).

The name of the message queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

message-queue-name: Specify the qualified name of the message queue that is used to receive messages.

JOBQ

Specifies the job queue on which the job entries are placed. A job entry is placed on this queue for each job in the input stream that has JOBQ(*RDR) specified on the Batch Job (BCHJOB) command. If *RDR is not specified on the BCHJOB command, the job queue specified on the BCHJOB command or in the job description is used. (The job queue for each job in the input stream can be different.) This parameter is valid only if ACTION(*SUBMIT) is specified on this command, in the existing network job entry, or in a subsequent Change Network Job Entry (CHGNETJOBE) command.

The name of the job queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

QBATCH: The job entry uses job queue QBATCH. The library list (*LIBL) is used to locate the named job queue; if one is found, the name of the library in which the job queue was found is placed in the job entry.

job-queue-name: Specify the name of the job queue.

Examples

Example 1: Submitting Input Streams Automatically

```
ADDNETJOBE FROMUSRID (JOHN SMITH)
  ACTION(*SUBMIT)
  SBMUSER(ANDERSON)
  JOBQ(QGPL/QPGMRL)
```

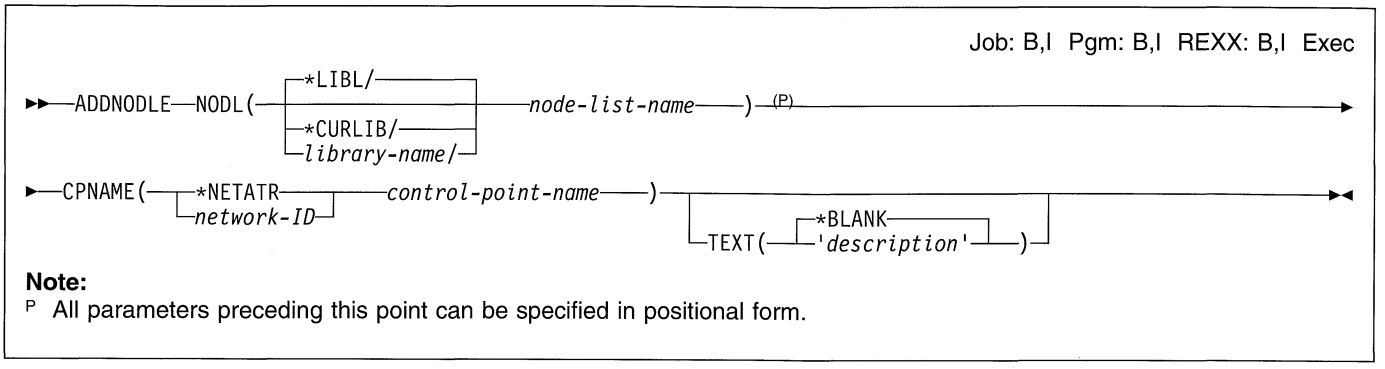
This command adds a network job entry that is used to determine the action that is taken for any input stream received from a user with a user ID of JOHN SMITH. The input streams are submitted automatically. User profile ANDERSON is used to check the authority to the job queues and job descriptions specified in the input stream. Messages are sent to the message queue specified in the user profile of the user to whom the input stream was sent. If no job queue is specified in either the received // BCHJOB command or the referenced job description, the jobs are placed on job queue QPGMRL in the QGPL library.

Example 2: Sending Messages to Specific Message Queue

```
ADDNETJOBE FROMUSRID(*ANY JONES) ACTION(*FILE)
  MSGQ(BROWN) SBMUSER(ANDERSON)
```

This command adds a network job entry that is used to determine the action taken for any input stream received from any user with JONES as the second part of the user ID for whom there is not a specific network job entry. The input stream is placed on the queue of received files for the user to whom the job was sent, and a message is sent to message queue BROWN.

ADDNODLE (Add Node List Entry) Command



Note:
P All parameters preceding this point can be specified in positional form.

Purpose

The Add Node List Entry (ADDNODLE) command adds a new entry to an existing node list object.

Required Parameters

NODL
Specifies the qualified name of the node list object to which the entry is added.

The name of the node list can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

node-list-name: Specify the name of the node list to which the entry is added.

CPNAME
Specifies the system to add to the node list object. The system is specified by its network ID and control point name.

***NETATR:** The NETID network attribute is used as the value of the network identifier (ID) of the system being added to the node list. If the NETID network attribute is changed, the new value does not affect the existing entries.

network-ID: Specify the network ID of the system being added to the node list.

control-point-name: Specify the control point name of the system being added to the node list.

Optional Parameter

TEXT
Specifies text that briefly describes the node list entry. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***BLANK:** Text is not specified.

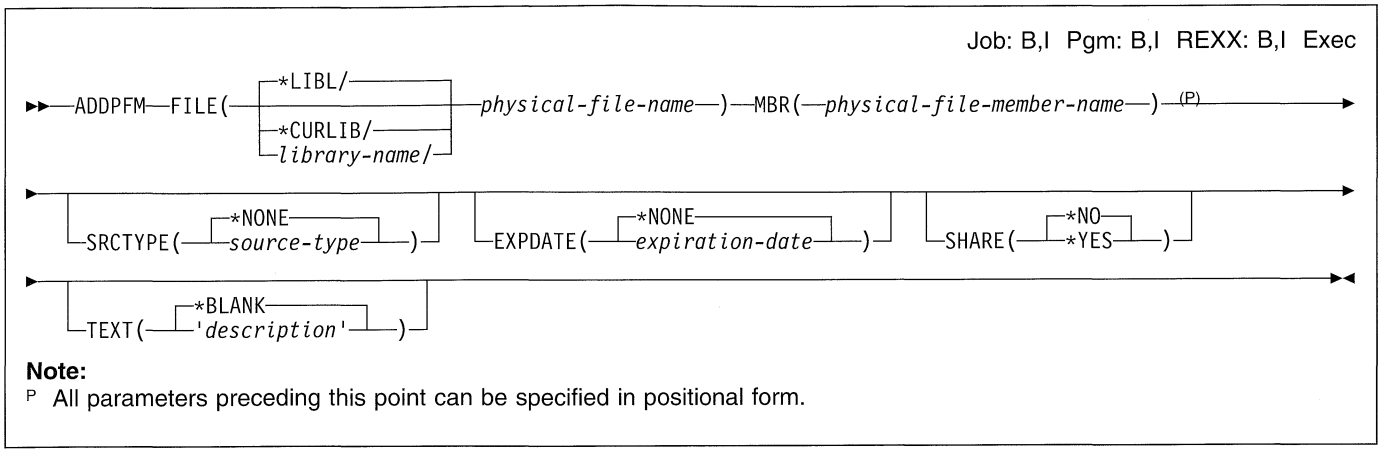
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
ADDNODLE   NODL(MYLIB/NODL02)
  CPNAME(*NETATR AS400A01)
  TEXT('The first AS/400 system in my network')
```

This command adds an entry for system AS400A01, which is in the local network, to the node list NODL02 in library MYLIB. The text description for the entry is 'The first AS/400 system in my network'.

ADDPFM (Add Physical File Member) Command



Note:

P All parameters preceding this point can be specified in positional form.

Purpose

The Add Physical File Member (ADDPFM) command adds a named member to the specified physical file, which must already exist on the system. A member must be added to the physical file before the file can have data stored in it. The first member of a file can be added by entering an ADDPFM command or by specifying a member name in the MBR parameter of the Create Physical File (CRTPF) command. To add other members to the file, use the ADDPFM command to specify each one.

The number of members that can be added to the physical file is limited to the number specified in the MAXMBRS parameter of the associated CRTPF command. Each member added has the same attributes as those defined in the physical file, its own set of data records, and its own access path, as specified in the data description specifications (DDS). The access path determines the order in which the records in that member are processed.

Note: An *EXCLRD lock is required on the file to add a member. Because this command adds a member to a file in a library, the library must not be locked (*SHRNUP or *EXCLRD in the Allocate Object command) for another job.

Required Parameters

FILE

Specifies the qualified name of the physical file to which this member is added.

The name of the physical file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

physical-file-name: Specify the name of the file to which the member is added.

If a DDM file is specified, the request is valid only to a target System/38 or AS/400 system.

MBR

Specifies the name of the physical file member being added. The member name must be unique in the file to which it is added.

If a DDM file is specified on the FILE parameter, and a member name is specified as part of the remote file name of the DDM file, the MBR name specified must match the member name in the remote file name in the DDM file.

Optional Parameters

SRCTYPE

Specifies the source type of a member if this is a source file. The source type option is a character string of no more than 10 characters representing a name. The first character must be alphabetic (including the characters $, @, or #), and the remaining characters must be alphanumeric or an underscore character.

Notes:

1. The user of this command must ensure the validity of the source type option.
2. The source type option can only be used with the Add Physical File Member (ADDPFM) command to add the source type attribute for a source file member.

***NONE:** No source type is specified.

source-type: Specify the source type of a member.

EXPDATE

Specifies the expiration date. The files cannot be overwritten until the expiration date. The expiration date must be later than or equal to the current date.

Note: An attempt to open a file member that has exceeded its expiration date causes an error message to be sent. (The RMVM command is used to remove the member from the file.)

***NONE:** No expiration date is specified.

expiration-date: Specify the date after which the member cannot be used. The expiration date must be specified in the format defined by the job attributes, DATFMT and DATSEP. The date must be enclosed in apostrophes if special characters are used in the format.

SHARE

Specifies whether the open data path (ODP) for the physical file member is shared with other programs in the routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

More information on shared database files is in the *Database Guide*.

***NO:** The ODP created by the program with this attribute is not shared with other programs in the routing step. Every time a program opens the file with this attribute, a new ODP to the file is created and activated.

***YES:** The ODP created with this attribute is shared with each program in the routing step that also specifies SHARE(*YES) when it opens the file.

Note: When SHARE(*YES) is specified and control is passed to a program, a read operation in that program retrieves the next input record. A write operation produces the next output record.

TEXT

Specifies text that briefly describes the physical file member. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***BLANK:** Text is not specified.

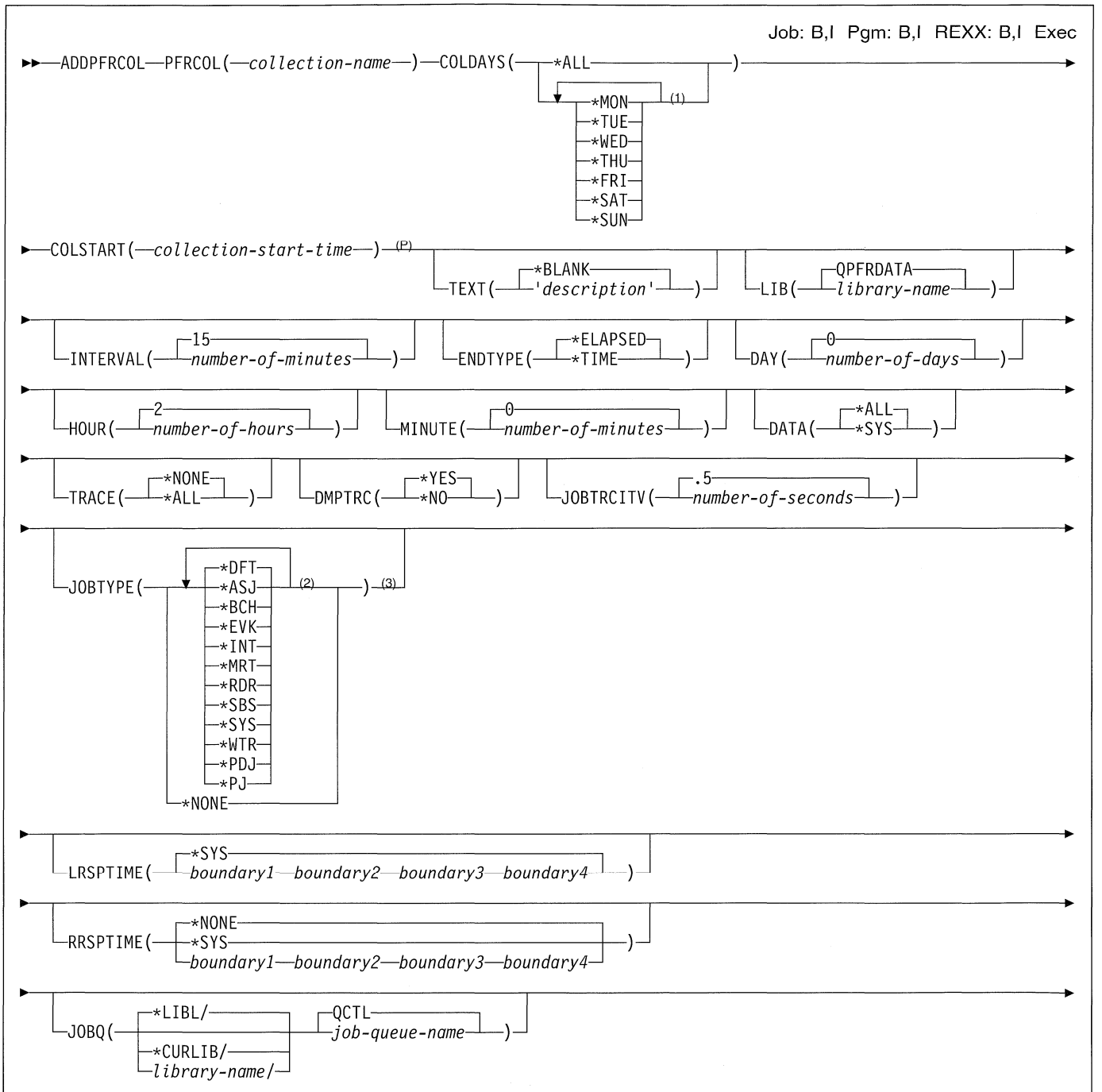
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
ADDPFM   FILE(INVENTX)  MBR(MONDAYTX)
  TEXT('Monday''s Inventory Transactions')
```

This command adds a member named MONDAYTX to the physical file named INVENTX. The library list (*LIBL) is used to find the file because the FILE value is not qualified by a library name. The size of the member and the storage allocation values assigned to this member were specified in the CRTPF command that created the physical file. The text, *Monday's Inventory Transactions*, describes this member of the INVENTX file.

ADDPFRCOL (Add Performance Collection) Command



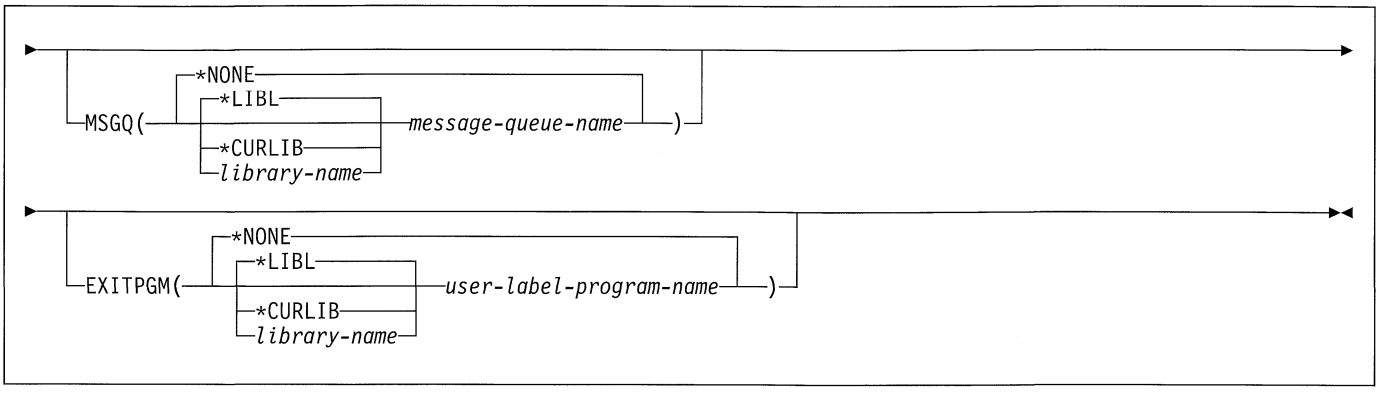
Notes:

1 A maximum of 7 repetitions

P All parameters preceding this point can be specified in positional form.

2 A maximum of 11 repetitions

3 Amaximum of 10 unique job types can be traced.



Purpose

The Add Performance Collection (ADDPFRCOL) command allows the user to add a performance collection. This is a weekly schedule that specifies when the Start Performance Monitor (STRPFRMON) command is sent. The collection consists of performance monitor information resulting from the weekly schedule that specifies when the STRPFRMON command is sent.

This command causes the STRPFRMON command to be sent on the days specified on the COLDAYS parameter and at the time specified on the COLSTART parameter.

Automatic performance collection requires a batch job, QPFRCOL, which queries the schedule created by the ADDPFRCOL command, and submits the STRPFRMON command at the appropriate times. If the user is using automatic performance collection for the first time, then the user needs to submit the performance collection job as follows:

```
SBMJOB   JOBD(QGPL/QPFRCOL)  USER(*JOBD)
  RQSDTA(*JOBD)  RTGDTA(*JOBD)
```

The user must submit this job after using the ADDPFRCOL command. This batch job runs until all the performance collections are removed or held. The job exists as an autostart job entry in the IBM-supplied subsystems, QBASE and QCTL. If the current release of one of these subsystems is being used, then as long as the performance collections are defined, the batch job will be started after each initial program load (IPL). If the current release of QBASE or QCTL is not being used, then the user can add an autostart job entry to the performance collection batch job on one of the user's subsystems. More information is in the *Work Management Guide*.

Required Parameters

PFRCOL

Specifies the name assigned to the performance collection.

COLDAYS

Specifies the days of the week on which the STRPFRMON command is sent.

***ALL:** The command is sent every day.

Days of the Week:

***MON:** The command is sent every Monday.

***TUE:** The command is sent every Tuesday.

***WED:** The command is sent every Wednesday.

***THU:** The command is sent every Thursday.

***FRI:** The command is sent every Friday.

***SAT:** The command is sent every Saturday.

***SUN:** The command is sent every Sunday.

COLSTART

Specifies the time of day in the format, hhmm, where hh is the hour and mm are the minutes, when the STRPFRMON command is sent. Specify the collection start time. The user can specify a value ranging from 0000 through 2359.

Optional Parameters

TEXT

Specifies text that briefly describes the performance collection. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

LIB

Specifies the library where the database files for the performance monitoring information is collected. Each file that is not found in the specified library is automatically created by the system in the library.

QPFRDATA: The data is located in the IBM-supplied performance data library, QPFRDATA.

library-name: Specify the name of the library to be searched.

INTERVAL

Specifies the time interval (in minutes) between each collection of system performance monitoring information.

15: A collection interval of 15 minutes is used.

number-of-minutes: Specify a collection interval ranging from 5 through 60 minutes.

ENDTYPE

Specifies how the performance monitor determines when to stop collecting information.

***ELAPSED:** Collection stops after the specified amount of time has elapsed as specified in the HOUR and MINUTE parameters. Changes to the system clock do not affect the stop time once the performance monitor has been started.

***TIME:** Collection stops on the day and time specified on the DAY, HOUR, and MINUTE parameters.

DAY

Specifies, if ENDTYPE(*ELAPSED) is specified, the number of days to collect performance monitoring information. Specifies, if ENDTYPE(*TIME) is specified, the number of days to collect performance monitoring information based on the start day until the collection ends.

0: Information is collected for 0 full days. Information can be collected for less than one full day using the HOUR and MINUTE parameters.

number-of-days: Specify the number of days to collect performance monitoring information. Valid values range from 0 through 7.

HOUR

Specifies, if ENDTYPE(*ELAPSED) is specified, the number of hours to collect information. Specifies, if ENDTYPE(*TIME) is specified, the hour of the day when the collection ends.

2: If ENDTYPE(*ELAPSED) is specified, the collection period is two hours. If ENDTYPE(*TIME) is specified, collection ends at 2:00 a.m.

number-of-hours: Specify the number of hours to collect information. If ENDTYPE(*ELAPSED) is specified, valid values range from 0 through 168. If ENDTYPE(*TIME) is specified, valid values range from 0 through 23.

MINUTE

Specifies, if ENDTYPE(*ELAPSED) is specified, the number of minutes during which performance monitoring information is collected. Specifies, if ENDTYPE(*TIME) is specified, the minute of the specified hour when the collection ends.

0: If ENDTYPE(*ELAPSED) is specified, information is collected for 0 minutes. If ENDTYPE(*TIME) is specified, collection ends on the hour.

number-of-minutes: Specify the number of minutes. Valid values range from 0 through 59.

DATA

Specifies the type of information collected.

***ALL:** All information, including system information and communications information, is collected.

***SYS:** Only system information is collected.

TRACE

Specifies the type of internal trace being started.

***NONE:** No trace is started.

***ALL:** All internal traces that contain performance related information are started.

DMPTRC

Specifies whether the trace is dumped when the collection of performance monitoring information ends.

***YES:** The trace is dumped when the collection ends.

***NO:** The trace is not dumped when the collection ends. The trace can be dumped at a later time by using the Dump Trace (DMPTRC) command.

JOBTRCITV

Specifies the time (in CPU seconds) between each collection of the job trace information.

.5: A time slice quantum interval of .5 seconds is used.

number-of-seconds: Specify a time slice quantum interval ranging from .5 through 9.9 seconds.

JOBTYPE

Specifies the type of job being traced. A maximum of 10 unique job types in addition to *DFT can be traced.

***DFT:** Batch and autostart jobs are traced.

***ASJ:** Autostart jobs are traced.

***BCH:** Batch jobs are traced.

***EVK:** Jobs started by a procedure start request are traced.

***INT:** Interactive job types are traced.

***MRT:** Multiple requester terminal jobs are traced.

***RDR:** Reader jobs are traced.

***SBS:** Subsystem monitor jobs are traced.

***SYS:** System jobs are traced.

***WTR:** Writer jobs are traced.

***PDJ:** Print driver jobs are traced.

***PJ:** Prestart jobs are traced.

Other Single Values

***NONE:** No jobs are traced.

LRSPTIME

Specifies the local work station response time categories. The performance monitor keeps track of interactive response times for each local work station attached to a controller that supports the collection of response time information. The response times are grouped into five categories; this parameter defines each category. Each value must contain a three-position decimal number, with one decimal position.

***SYS:** The system response categories are:

- 0-1 seconds
- 1-2 seconds

- 2-4 seconds
- 4-8 seconds
- Longer than 8 seconds

Element 1: First Boundary

boundary1: Specify the first response time boundary. All responses falling between zero and this boundary value are counted in the first response time category. This is a DEC(3,1) variable.

Element 2: Second Boundary

boundary2: Specify the second response time boundary. All responses falling between the first boundary value and this boundary value are counted in the second response time category. This is a DEC(3,1) variable.

Element 3: Third Boundary

boundary3: Specify the third response time boundary. All responses falling between the second boundary value and this boundary value are counted in the third response time category. This is a DEC(3,1) variable.

Element 4: Fourth Boundary

boundary4: Specify the fourth response time boundary. All responses falling between the third boundary value and this boundary value are counted in the fourth response time category. This is a DEC(3,1) variable. All response times greater than this value are counted in the fifth response time category.

RRSPTIME

Specifies the remote work station response time categories. The performance monitor keeps track of interactive response times for each remote work station attached to a controller that supports the collection of response time information. The response times are grouped into five categories; this parameter defines each category. Each value must contain a three-position decimal number, with one decimal position.

***NONE:** Remote work station response time is not collected.

***SYS:** The system response categories are:

- 0-1 seconds
- 1-2 seconds
- 2-4 seconds
- 4-8 seconds
- Longer than 8 seconds

Element 1: First Boundary

boundary1: Specify the first response time boundary. All responses falling between zero and this boundary value are counted in the first response time category. This is a DEC(3,1) variable.

Element 2: Second Boundary

boundary2: Specify the second response time boundary. All responses falling between the first boundary value and this boundary value are counted in the second response time category. This is a DEC(3,1) variable.

Element 3: Third Boundary

boundary3: Specify the third response time boundary. All responses falling between the second boundary value and this boundary value are counted in the third response time category. This is a DEC(3,1) variable.

Element 4: Fourth Boundary

boundary4: Specify the fourth response time boundary. All responses falling between the third boundary value and this boundary value are counted in the fourth response time category. This is a DEC(3,1) variable. All response times greater than this value are counted in the fifth response time category.

JOBQ

Specifies the qualified name of the job queue on which this job is placed.

QCTL: The IBM-supplied controlling subsystem, QCTL, is used.

The name of the job queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

job-queue-name: Specify the name of the job queue.

MSGQ

Specifies the qualified name of the message queue to which messages are sent.

Note: The performance monitor can send its messages to the user-defined message queue and the system operator message queue (QSYS/QSYSOPR). Messages sent to the user-defined message queue communicate status information relating to the performance monitoring of startup and end operations, as well as other general system problems.

***NONE:** Messages are sent to the system operator message queue only.

The name of the message queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

message-queue-name: Specify the name of the message queue where messages are sent.

EXITPGM

Specifies the user-written exit program that is called to process the performance data collected by this command.

***NONE:** No exit program is specified.

The name of the exit program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

user-label-program-name: Specify the user label name of the exit program to be used.

Examples

Example 1: Using STRPFRMON Command Defaults

```
ADDPFRCOL  PFRCOL(MONDAYS)
  COLDAYS(*MON)
  COLSTART(0800)
  TEXT('Mondays 8:00 to 10:00')
```

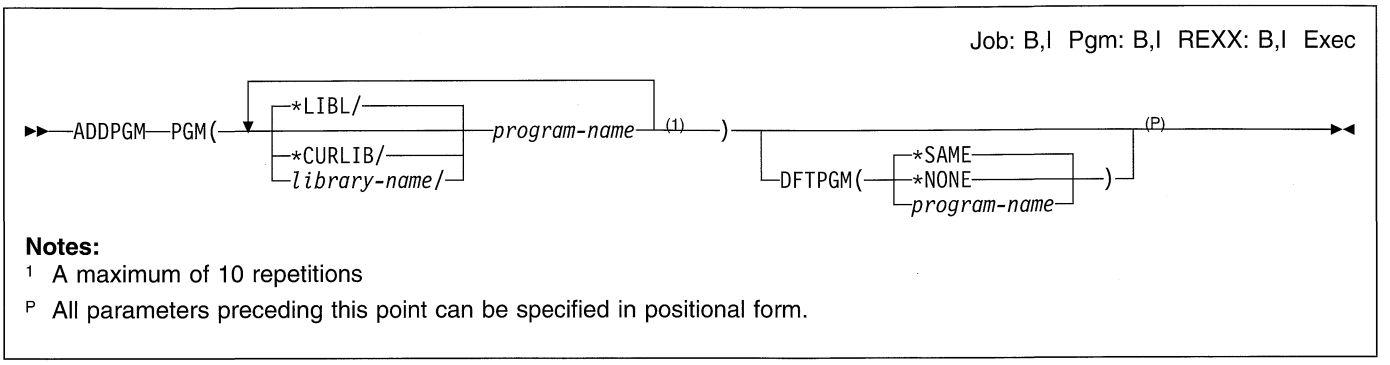
This command defines the collection named MONDAYS. The Start Performance Monitor (STRPFRMON) command is sent every Monday at 8:00 a.m. The command defaults are used so that the monitor collects information during an elapsed time of 2 hours.

Example 2: Specifying Collection Time

```
ADDPFRCOL  PFRCOL(AFTERNOON)
  COLDAYS(*MON *TUE *WED *THU *FRI)
  COLSTART(1330)  ENDTYPE(*TIME)
  HOUR(16)  MINUTE(30)
```

This command defines the collection named AFTERNOON. The Start Performance Monitor (STRPFRMON) command is sent every day, Monday through Friday, at 1:30 p.m. The STRPFRMON command is changed so that the monitor collects information for 3 hours ending at 4:30 p.m.

ADDPGM (Add Program) Command



Purpose

The Add Program (ADDPGM) command adds 1 to 10 programs to the group of programs currently being debugged. When included in debug mode, the specified programs can have breakpoints and traces added to them for controlling and tracing their processing. The values of the programs' variables can also be shown and changed.

When debugging one job from another job, debugging affects the running of the programs in the job being debugged, but not in the job doing the debugging. The user may run programs in a job doing the debugging, however, the programs will not be debugged.

Restrictions:

1. No more than 10 programs can be debugged at the same time.
2. Two or more programs with the same name cannot be debugged at the same time.
3. This command is valid only in debug mode. To start debug mode, see the Start Debug (STRDBG) command.
4. This command cannot be used if the user is servicing another job, and that job is on a job queue, or is being held, suspended, or ended.
5. This command cannot be used to add bound programs.
6. The user must have either *CHANGE authority to the program, or *USE authority to the program and *SERVICE special authority.

Required Parameter

PGM

Specifies the qualified name of one or more programs being debugged. The number of programs specified here depends on how many programs are already being debugged; 10 is the maximum at any time. The user cannot debug two programs that have the same name.

The name of the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the program to be added for debugging.

Optional Parameter

DFTPGM

Specifies the name of the program to use as the default program during debug mode. The program specified here is used as the default program for any of the other debug commands that specify *DFTPGM on their PGM parameter. That is, if a default program was previously specified, this parameter can change it.

***SAME:** The value does not change.

***NONE:** No program is specified as the default program; if a program was specified as a default program, it is no longer the default program. If the job has no default program, *DFTPGM cannot be specified on the PGM parameter of any other debug commands.

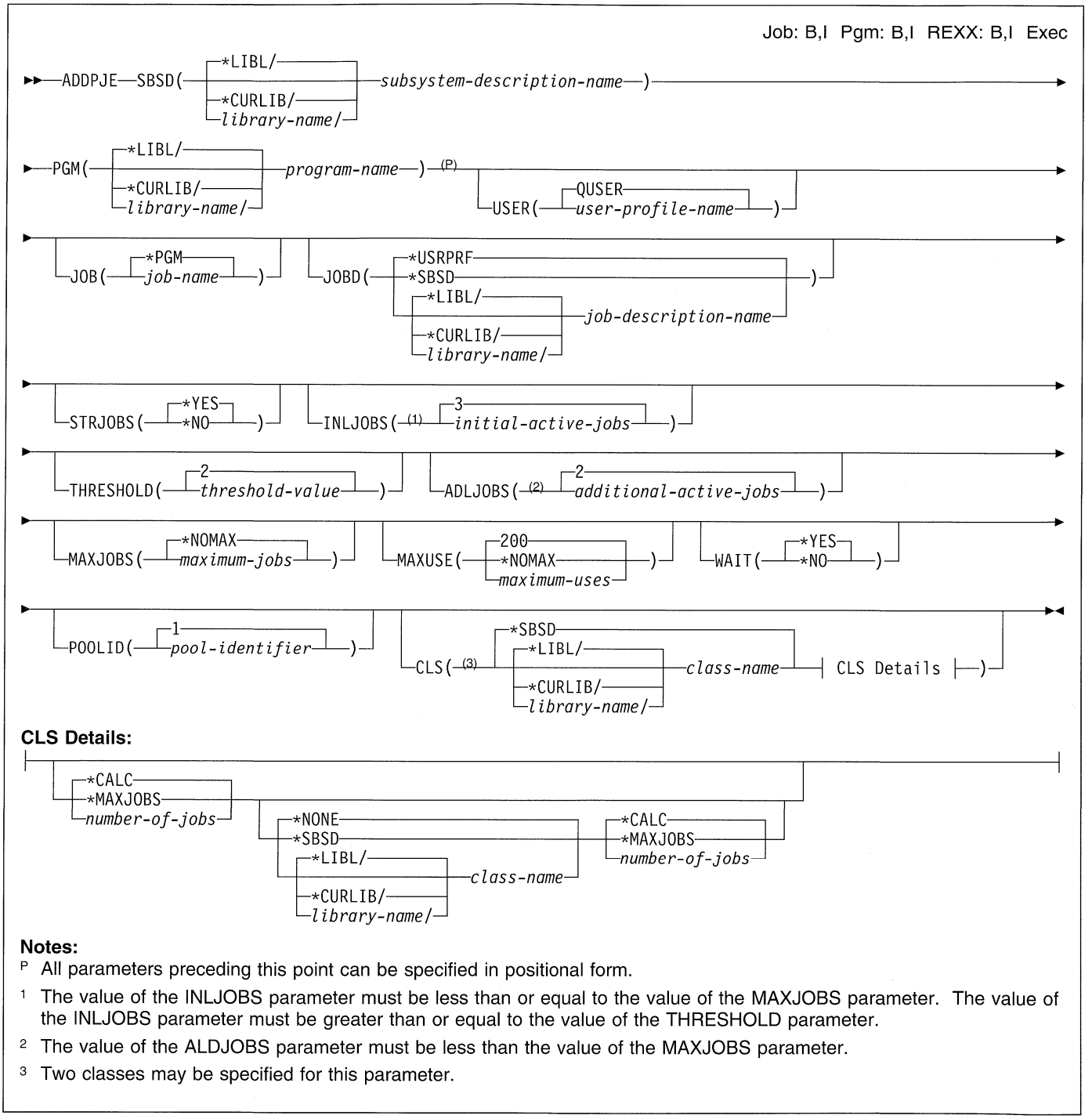
program-name: Specify the name of the program to use as the default program during debug mode. The same name must also be specified in the PGM parameter of this command or have been specified on the Start Debug (STRDBG) command or on a previous Add Program (ADDPGM) command.

Example

```
ADDPGM    PGM(QGPL/MYPROG)
```

This command adds the program MYPROG, located in the QGPL library, to the current debug mode. Breakpoints and traces can be put in MYPROG, and its variables can be displayed and changed by other debug commands. Because DFTPGM was not specified, the default program is not changed.

ADDPJE (Add Prestart Job Entry) Command



Notes:

P All parameters preceding this point can be specified in positional form.

1 The value of the INLJOBS parameter must be less than or equal to the value of the MAXJOBS parameter. The value of the INLJOBS parameter must be greater than or equal to the value of the THRESHOLD parameter.

2 The value of the ALDJOBS parameter must be less than the value of the MAXJOBS parameter.

3 Two classes may be specified for this parameter.

Purpose

The Add Prestart Job Entry (ADDPJE) command adds a prestart job entry to the specified subsystem description. The associated subsystem must be inactive when the prestart job entry is added. The entry identifies prestart jobs that are started when the subsystem is started or when the start prestart job command is entered.

Restriction: This command is restricted to a user with *USE and object management authorities for the subsystem description and *USE authority for the user profile and the job description.

Required Parameters

SBSD

Specifies the qualified name of the subsystem description to which the prestart job entry is being added.

The name of the subsystem description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

subsystem-description-name: Specify the name of the subsystem description.

PGM

Specifies the qualified name of the program run by the prestart job. This program name is used to match an incoming program start request with an available prestart job. Two entries with the same program name can exist in a single subsystem description, but they must have different library names. If the program does not exist when the entry is added, a library qualifier must be specified because the qualified name is retained in the subsystem description.

The name of the the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the program to be run by the prestart job.

Optional Parameters

USER

Specifies the name of the user profile under which the prestart job runs when it is not handling a program start request.

QUSER: The IBM-supplied QUSER user profile is used.

user-profile-name: Specify the name of the user profile used for the prestart job.

JOB

Specifies the name of the prestart job that is started.

***PGM:** The prestart job name is the same as the program name specified by the PGM parameter.

job-name: Specify the name of the prestart job.

JOBD

Specifies the name of the job description used. If the job description does not exist when the entry is changed, a library qualifier must be specified because the qualified job description name is retained in the subsystem description. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***USRPRF:** The job description name specified in the user profile for the USER parameter is used.

***SBSD:** The job description having the same name as that of the subsystem description named in the SBSD parameter is used.

The name of the job description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

job-description-name: Specify the name of the job description being used for this prestart job. If no library is specified, the library list (*LIBL) of the job where this command is run is used to find the job description.

STRJOBS

Specifies whether prestart jobs are started at the time the subsystem is started.

***YES:** The prestart jobs are started when the subsystem is started.

***NO:** The prestart jobs are not started when the subsystem is started. The Start Prestart Jobs (STRPJ) command must be used to start these prestart jobs.

INLJOBS

Specifies the initial number of prestart jobs started when the subsystem specified in the SBSD parameter is started.

Notes:

1. The value of this parameter must be less than or equal to the value of the MAXJOBS parameter.
2. The value of this parameter must be greater than or equal to the value of the THRESHOLD parameter.

3: Three prestart jobs are started when the subsystem is started.

initial-active-jobs: Specify the initial number of prestart jobs that are started when the subsystem is started. Valid values range from 1 through 1000.

THRESHOLD

Specifies the threshold number of available prestart jobs at which additional prestart jobs are started. When the

pool of available prestart jobs (jobs available to service program start requests) is reduced below this number, more prestart jobs (specified by the ADLJOBS parameter) are started and added to the pool of available prestart jobs.

Note: The value of this parameter must be less than or equal to the value of the INLJOBS parameter.

2: When one prestart job is available, start the number of jobs specified by the ADLJOBS parameter.

threshold-value: Specify the minimum number of prestart jobs that must be available before additional prestart jobs are started. Valid values range from 1 through 1000.

ADLJOBS

Specifies the additional number of prestart jobs started when the number of prestart jobs drops below the THRESHOLD parameter.

Note: The value of this parameter must be less than the value of the MAXJOBS parameter.

2: Two additional prestart jobs are started.

additional-active-jobs: Specify the number of additional prestart jobs to be started. Valid values range from 0 through 1000.

MAXJOBS

Specifies the maximum number of prestart jobs that can be active on the subsystem at the same time under this prestart job entry.

Notes:

1. The value of this parameter must be greater than or equal to the value of the INLJOBS parameter.
2. The value of this parameter must be greater than the value of the ADLJOBS parameter.

***NOMAX:** There is no maximum number of jobs that can be active at the same time.

maximum-jobs: Specify the maximum number of prestart jobs that can be active at the same time. Valid values range from 1 through 1000.

MAXUSE

Specifies the maximum number of program start requests that can be handled by each prestart job in the pool before the job controlled by the system is ended.

200: A prestart job for this entry can service up to 200 program start requests before it is ended. If *NOMAX is specified, the prestart jobs may end abnormally for one of the following reasons:

- The job log has exceeded the maximum size.
- The allowed maximum for spooled files has been reached.
- The allowed maximum for CPU time has been reached.
- The allowed maximum for temporary storage has been reached.

***NOMAX:** There is no maximum number of jobs that can be active at the same time.

maximum-uses: Specify the maximum number of program start requests that a prestart job can handle before it is ended. Valid values range from 1 through 1000.

WAIT

Specifies whether program start requests either wait for a prestart job to become available or are rejected if a prestart job is not immediately available when the program start request is received.

Note: Refer to the *CL Programmer's Guide* to determine the time-out considerations for the communications type being used.

***YES:** Program start requests wait either until there is an available prestart job, or until a prestart job is started, to service the request.

***NO:** Program start requests are rejected if a prestart job is not immediately available when the program start request is received.

POOLID

Specifies the identifier of the subsystem pool in which the prestart jobs run.

1: The prestart jobs run in the first pool.

pool-identifier: Specify the identifier of the subsystem pool in which the prestart jobs run. Valid values range from 1 through 10.

CLS

Specifies the names of the classes under which the prestart jobs run and how many prestart jobs are allowed to run under each class. Jobs start by using the first class. After the number of jobs specified for the first class is reached, jobs are started under the second class. If the class does not exist when the entry is added, a library qualifier must be specified because the qualified class name is retained in the subsystem description.

Element 1: Class Name of First Class

***SBSD:** The class having the same name as the subsystem description specified in the SBSD parameter is used for prestart jobs.

The name of the class can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

class-name: Specify the name of the class being used for prestart jobs. If no library qualifier is specified, the

library list (*LIBL) of the job where this command is run is used to find the class.

Element 2: Number of Jobs Using First Class

***CALC:** The system calculates how many prestart jobs use this class. If one class is specified and *CALC is specified, all of the jobs use the specified class. If two classes are specified and *CALC is specified for both, the first class is the value of the MAXJOBS parameter divided by two, and the second class is the value of the MAXJOBS parameter minus the value calculated for the first class. If a specific number of jobs is specified for one class and *CALC is specified for the other class, the system calculates the difference between MAXJOBS and the specific number of jobs for the *CALC designation.

***MAXJOBS:** All of the prestart jobs use the specified class.

number-of-jobs: Specify the number of prestart jobs that use this class. The sum of the values specified for both classes must total the value specified for the MAXJOBS parameter.

Element 3: Class Name of Second Class

***NONE:** One class is used.

***SBSD:** The class having the same name as the subsystem description specified in the SBSD parameter is used for prestart jobs.

The name of the class can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

class-name: Specify the name of the class being used for prestart jobs. If no library qualifier is specified, the library list (*LIBL) of the job where this command is run is used to find the class.

Element 4: Number of Jobs Using Class

***CALC:** The system calculates how many prestart jobs use this class. If one class is specified and *CALC is specified, all of the jobs use the specified class. If two classes are specified and *CALC is specified for both, the first class is the value of the MAXJOBS parameter divided by two, and the second class is the value of the MAXJOBS parameter minus the value calculated for the first class. If a specific number of jobs is specified for one class and *CALC is specified for the other class, the system calculates the difference between MAXJOBS and the specific number of jobs for the *CALC designation.

***MAXJOBS:** All of the prestart jobs use the specified class.

number-of-jobs: Specify the number of jobs that use this class. The sum of the values specified for both classes must total the value specified for the MAXJOBS parameter.

Examples

Example 1: Specifying Additional Prestart Jobs

```
ADDPJE  SBSD(QGPL/PJSBS)  PGM(QGPL/PGM1)  INLJOBS(15)
  THRESHOLD(5)  ADLJOBS(10)  WAIT(*NO)
```

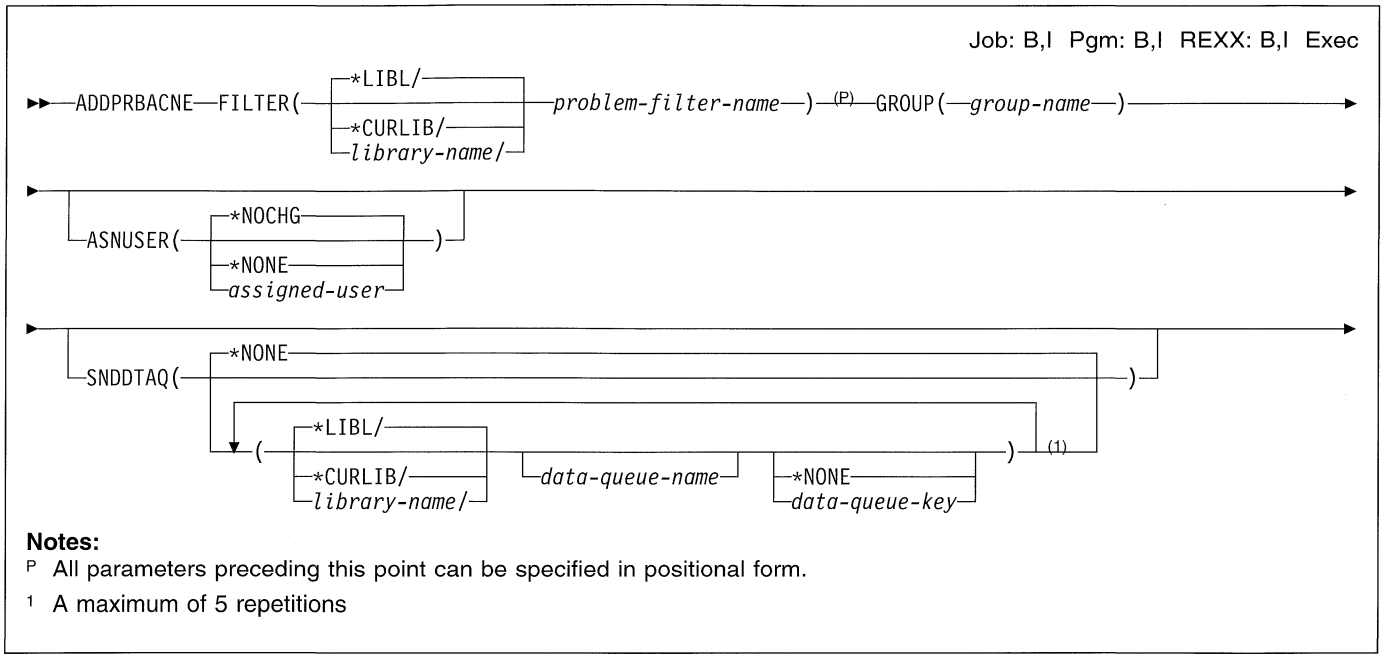
This command adds a prestart job entry for the PGM1 program in the QGPL library to the PJSBS subsystem description contained in the QGPL library. The entry specifies that 15 prestart jobs (program PGM1 in the QGPL library) are started when subsystem PJSBS in the QGPL library is started. When the pool of available prestart jobs is reduced to four (because the prestart jobs are servicing program start requests specified for program PGM1 in the QGPL library), ten additional jobs are started. If no prestart jobs are available for this entry when a program start request is received, the program start request is rejected.

Example 2: Specifying Maximum Number of Prestart Jobs

```
ADDPJE  SBSD(QGPL/PJSBS)  PGM(QGPL/PGM2)
  USER(PJUSER)  MAXJOBS(100)
  MAXUSE(50)  CLS(QGPL/CLS1 75 QGPL/CLS2 *CALC)
```

This command adds a prestart job entry for the PGM2 program in the QGPL library to the PJSBS subsystem description contained in the QGPL library. The entry specifies that the prestart job for this entry runs under the PJUSER user profile. The maximum number of prestart jobs that can be active at the same time for this entry is 100. Each prestart job in the pool can handle 50 program start requests before the job is ended. If 100 prestart jobs are active at the same time for this entry, 75 of them would use CLS1 in the QGPL library, and 25 of them would use CLS2 in the QGPL library. If 50 prestart jobs are active at the same time for this entry, all 50 of them would use class CLS1 in the QGPL library.

ADDPRBACNE (Add Problem Action Entry) Command



Notes:

| P All parameters preceding this point can be specified in positional form.

1 A maximum of 5 repetitions

Purpose

The Add Problem Action Entry (ADDPRBACNE) command adds an entry to the specified problem filter. This entry describes the actions to take for a problem entry. A problem entry is assigned to the specified group by a selection entry in the specified problem filter.

Required Parameters

FILTER

Specifies the name of the filter.

| The name of the filter can be qualified by one of the following library values:

*LIBL: All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

problem-filter-name: Specify the name of the filter.

GROUP

Specifies the group for which the actions are applied. The group name is assigned from selection criteria from a selection entry in the filter. Selection entries are added to the filter with the ADDPRBSLTE command.

Optional Parameters

ASNUSER

Specifies the user assigned to the problem log entry.

*NOCHG: No new value is assigned to the problem log entry.

***NONE:** No user is assigned to the problem log entry.

assigned-user: Specify a user name.

SNDDTAQ

Specifies the data queue for the problem notification record. Keyed data queues are supported.

*NONE: No data queue is used.

Element 1: Data Queue Name

| The name of the data queue can be qualified by one of the following library values:

*LIBL: All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

data-queue-name: Specify the name of the data queue.

Element 2: Data Queue Key

***NONE:** No key is used on the data queue.

data-queue-key: Specify the data queue key.

Notes:

1. If an exact match is not found for the group name when a filter is applied, then the default action entry is used to assign actions. The default action entry is automatically added to the filter when it is created. The default values are ASNUSER(*NONE) and SNDDTAQ(*NONE).
2. A keyed data queue is a queue with a key assigned to each entry on the queue. When retrieving entries, a key can be specified and the entries with that key are retrieved on a FIFO order. The key that is specified on the *SNDDTAQ parameter is assigned to the problem notification record when placed on a keyed data queue.

An 80-byte record is enqueued on the data queue specified by the user. This record is received when the QRCVDTAQ program is called. The data queue does not have to be used solely for problems; alerts and problems can share the same data queue.

If a key is specified, it is used when enqueueing the record on the queue. If the data queue is non-keyed, the record is enqueued without a key.

Note: The time stamp used is the system standard time stamp. This time is already stored in the problem record.

The following table describes the record format.

Table 6. Record Format

Position	Type	Value	Description
1-10	CHAR	*PRBFTR	Problem filtering notification
11-11	CHAR	Function	Function performed 1 - Problem created 2 - Problem changed 3 - Problem deleted
12-19	CHAR	Function TOD	TOD time stamp for function
20-29	CHAR	Group	Group problem was filtered into
30-39	CHAR	Problem ID	Problem ID number
40-59	CHAR	Origin System	System where problem originated
60-60	CHAR	Last Event	Last event committed into the history log (see note)
61-68	CHAR	Event TOD	TOD time stamp for Last Event
69-80	CHAR	Reserved	Reserved for future use

Note: Valid Last Event values are the following:

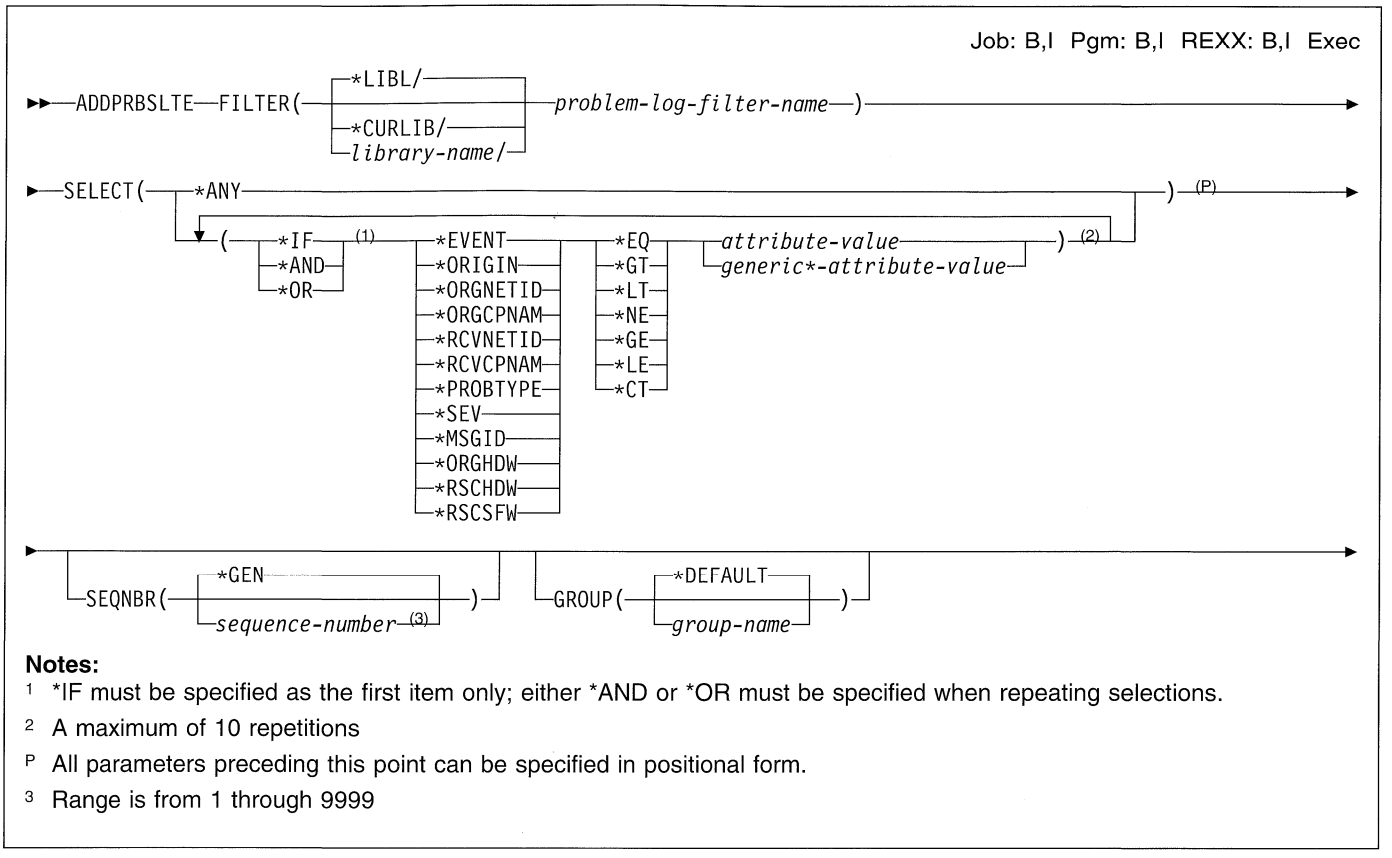
'01'X	Problem entry opened
'02'X	Request received
'03'X	Opened by Alert
'10'X	Problem analyzed
'11'X	Verification test ran
'12'X	Recovery procedure ran
'20'X	Prepared to report
'21'X	Service request sent
'22'X	Problem answered
'23'X	Response sent
'24'X	Reported by voice
'25'X	Fixes transmitted
'30'X	Fix verified
'41'X	Analyzed remotely
'42'X	Remote verification ran
'43'X	Remote recovery ran
'50'X	Alert created
'51'X	APAR created
'52'X	APAR data saved
'54'X	APAR data restored
'55'X	APAR data deleted
'60'X	Problem changed by Change Problem (CHGPRB) command
'61'X	Problem deleted by Delete Problem (DLTPRB) command
'99'X	Problem entry closed

Example

```
ADDPRBACNE   FILTER(MYLIB/MYFILTER)  GROUP(IOWA)
  ASNUSER(SYSOPR))   SNDDTAQ(*LIBL/PROBDTAQ)
```

The actions defined for group IOWA are: enqueue the problem on data queue PROBDTAQ; and assign the problem to user SYSOPR.

ADDPRBSLTE (Add Problem Selection Entry) Command



Notes:

1 *IF must be specified as the first item only; either *AND or *OR must be specified when repeating selections.

2 A maximum of 10 repetitions

P All parameters preceding this point can be specified in positional form.

3 Range is from 1 through 9999

Purpose

The Add Problem Selection Entry (ADDPRBSLTE) command allows you to define selection criteria that categorize a group of problem log entries. You can add a problem log selection entry to a problem log filter that was created using the Create Filter (CRTFTR) command.

Required Parameters

FILTER

Specifies the name of the filter.

The name of the filter can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

problem-log-filter-name: Specify the name of the filter.

SELECT

Specifies that a problem log entry is selected or not selected based on whether information in the problem log entry satisfies a specified relationship.

You can specify a single value (*ANY) or all four elements that define a relationship. When you specify the four elements, the attribute and attribute value are compared for the relationship specified by the relational operator.

***ANY:** Any problem log entry is selected.

Element 1: Logical Operator

***IF:** The specified relationship must be satisfied for a problem log entry to be selected.

***AND:** The specified relationship must be satisfied in addition to the *IF relationship for a problem log entry to be selected.

***OR:** The specified relationship must be satisfied in addition to or instead of the *IF relationship for a problem log entry to be selected.

Element 2: Attribute

***EVENT:** The filter is applied when the problem log entry is created (a value of 1), changed (a value of 2), or deleted (a value of 3). If the entry has been created and is changed before being committed, use the value of 1.

***ORIGIN:** The problem log entry was locally generated (a value of L) or was received from another system (a value of R).

***ORGNETID:** The network identifier (ID) of the system in which the problem log entry originated is specified. This information is displayed using the Work with Problems (WRKPRB) command which shows the details for a specific problem. Specify the value in the following form:

```
'nnnnnnnnnn'
```

***ORGCPNAM:** The control point name of the system in which the problem log entry originated is specified. This information is displayed using the Work with Problems (WRKPRB) command which shows the details for a specific problem. Specify the value in the following form:

```
'cccccccccc'
```

***RCVNETID:** The network identifier of the remote system from which the problem log entry was received is specified. This information is displayed using the Work with Problems (WRKPRB) command which shows the details for a specific problem. Specify the value on the following form:

```
'nnnnnnnnnn'
```

***RCVCPNAM:** This attribute specifies the Remote System Control Point name in which the problem log entry received from. This information is displayed using the Work with Problems (WRKPRB) command and shows the details for a specific problem. The value specified for this attribute should be of the following form:

```
'cccccccccc'
```

***PROBTYPE:** The type of problem entry created. Possible problems are machine detected (a value of 1), user detected (a value of 2), PTF order (a value of 3), or application detected (a value of 4).

Note: User-Detected Remote Hardware problems are grouped with number 2 User-Detected problems.

***SEV:** The severity of the problem log entry created. Possible choices are high (a value of 1), medium (a value of 2), low (a value of 3), none (a value of 4), or not assigned (a value of 5).

Note: Problems do not have a severity level when locally created.

***MSGID:** The message ID found in the problem log entry. This is usually an AS/400 message ID from an AS/400 system.

***ORGHDW:** The origin hardware resource information in the problem log entry. This information is displayed using the Work with Problems (WRKPRB) command and shows the details for a specific problem. Specify the value in the following form:

```
'tttt mmm ss-ssssssss'
'tttt mmm ss-sssss'
'tttt mmm sssssss'
'tttt mmm sssss'
```

where tttt is the machine type, mmm is the model number and sssssssss is the serial number. Use this exact format to match a particular hardware resource exactly, or use a part of the hardware value with the Contains (*CT) relation to provide a partial match.

***RSCHDW:** The failing hardware resource information in the problem log entry. This information is displayed using the Work with Problems (WRKPRB) command and shows the details for a specific problem. Specify the value in the following form:

```
'tttt mmm ss-ssssssss'
'tttt mmm ss-sssss'
'tttt mmm sssssss'
'tttt mmm sssss'
```

where tttt is the machine type, mmm is the model number and sssssssss is the serial number. Use this exact format to match a particular hardware resource exactly, or use a part of the hardware value with the Contains (*CT) relation to provide a partial match.

***RSCSFW:** The failing software resource information in the problem log entry. This information is displayed using the Work with Problems (WRKPRB) command and shows the details for a specific problem. Specify the value in the following form:

```
'ppppppp vv rr mm'
```

where ppppppp is the licensed program ID, vv is the version number, rr is the release number, and mm is the modification level. Use this exact format to match a particular software resource exactly, or use a part of the software value with the Contains (*CT) relation to provide a partial match.

Element 3: Relational Operator

The value specified for Element 2 must have the following relationship to Element 4:

*EQ	Equal to
*GT	Greater than
*LT	Less than
*NE	Not equal to
*GE	Greater than or equal to
*LE	Less than or equal to
*CT	Contains

Element 4: Attribute Value

attribute-value: Specify a value to compare with the contents of the attribute specified for Element 2. A maximum of 30 characters can be specified. The value must be specified in character format and must be enclosed in apostrophes if it contains blanks or special characters. If a CL variable is specified for the value, it must be a character variable.

generic-attribute-value:* Specify the generic attribute value. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has

authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be added only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

Optional Parameters

SEQNBR

Specifies the sequence number of the problem log selection entry. Selection entries in a filter are numbered by sequence number. When a filter is applied, the selection entries are tested in order of ascending sequence number.

***GEN:** The system generates the sequence number.

sequence-number: Specify a number from 1 through 9999.

GROUP

Specifies the group to which a problem log entry is assigned if it matches the criteria specified on the SELECT parameter.

***DEFAULT:** The problem log entry is assigned to the default group.

group-name: Specify a group name.

Examples

Example 1: Adding a Selection Entry

```
ADDPRBSLTE
   FILTER(PROBLIB/PROBFILTER)  +
   SELECT((*IF *EVENT  *EQ 1)  +
          (*AND *SEV   *EQ 1)) +
   SEQNBR(*GEN)                +
   GROUP(HIGHPROB)
```

This command adds an entry to the filter PROBFILTER in library PROBLIB. Any problems that have been created and are of severity 1 are assigned to group HIGHPROB.

Example 2: Assigning Entries by Origin System Network ID

```
ADDPRBSLTE
   FILTER(PROBLIB/PROBFILTER)          +
   SELECT((*IF *ORGNETID *EQ 'IOWA')) +
   SEQNBR(*GEN)                        +
   GROUP(IOWA)
```

This command assigns any problems with a origin system network ID of IOWA to group IOWA.

Example 3: Assigning Entries by Problems for Messages

```
ADDPRBSLTE
   FILTER(PROBLIB/PROBFILTER)          +
   SELECT((*IF *MSGID *EQ 'CPF89*')) +
   SEQNBR(*GEN)                        +
   GROUP(MSGCPF89)
```

This command assigns any problems for message CPF8901, CPF8902, and so on, to group MSGCPF89.

Example 4: Assigning Entries by Hardware Problems

```
ADDPRBSLTE
   FILTER(PROBLIB/PROBFILTER)       +
   SELECT((*IF  *RSCHDW *CT 9404)   +
          (*OR  *RSCHDW *CT 9406)   +
          (*OR  *RSCHDW *CT 9402)) +
   SEQNBR(*GEN)                     +
   GROUP(AS400USER)
```

All problems for AS/400 hardware (the hardware resource information *containing* machine type 9402, 9404 or 9406) are assigned to group AS400USER.

Caution must be taken when using the contains operation. In this example if the sending machine had a serial number containing 9402, 9404, or 9406 it would also match this selection entry even if the machine type was not 9402, 9404, or 9406. A better example follows.

Example 5: Assigning Entries by Hardware Problems

```
ADDPRBSLTE
   FILTER(PROBLIB/PROBFILTER)       +
   SELECT((*IF  *RSCHDW *EQ 9404*)  +
          (*OR  *RSCHDW *EQ 9406*)  +
          (*OR  *RSCHDW *EQ 9402*)) +
   SEQNBR(*GEN)                     +
   GROUP(AS400USER)
```

This command assigns all problems for AS/400 hardware (the hardware resource information *equals* machine type 9402, 9404 or 9406) to group AS400USER.

This is a better way to select on the sending hardware machine type. Only those machines with types of 9402, 9404, or 9406 will result in a match.

Example 6: Assigning Entries by Machine-detected Problems

```
ADDPRBSLTE
   FILTER(PROBLIB/PROBFILTER)       +
   SELECT((*IF  *PROBTYPE *EQ 1))   +
   SEQNBR(*GEN)                     +
   GROUP(MACHDETECT)
```

This command assigns any problems that are machine-detected to group MACHDETECT.

Example 7: Assigning Entries by Product-specific Problems

```
ADDPRBSLTE
   FILTER(PROBLIB/PROBFILTER)                +
   SELECT((*IF *RSCSFW *EQ '5738SS1 02 01 00')) +
   SEQNBR(15)                                +
   GROUP(OS400V2R1)
```

This command assigns any problems that are specifically for OS/400 Version 2 Release 1 modification level 0 to group OS400V2R1. Notice that this entry is placed after entry number 10 in the filter, since 15 is specified as the sequence number.

Example 8: Assigning Entries by Matching Products

```
ADDPRBSLTE
   FILTER(PROBLIB/PROBFILTER)            +
   SELECT((*IF *RSCSFW *EQ '5738SS1*')) +
   SEQNBR(25)                            +
   GROUP(OS400)
```

This selection entry matches any Version 2 release of OS/400.

Notes:

1. The order of selection entries within a filter is important. When the filter is applied to the problem log entry, the selection entries are examined from the *first* entry to the *last* entry in ascending order. The *first* selection entry that matches a problem is used. To ensure correct operation the most specific selection entries should be first, and the least specific selection entries last.
2. If the selection entries are not order specific (i.e. each selection entry matches one and only one problem) then the most likely or the most common should be placed first. This will ensure the best performance as fewer selection entries will need to be checked.
3. If no selection entries result in a match when a filter is applied, then the *LAST selection entry is used to assign a group. The *LAST selection entry is automatically added to the filter when it is created. The SELECT parameter for the *LAST selection entry is *ANY, which will always result in a match.
4. The *AND logical operator takes precedence over the *OR logical operator within a selection entry. Therefore, the following SELECT specification:

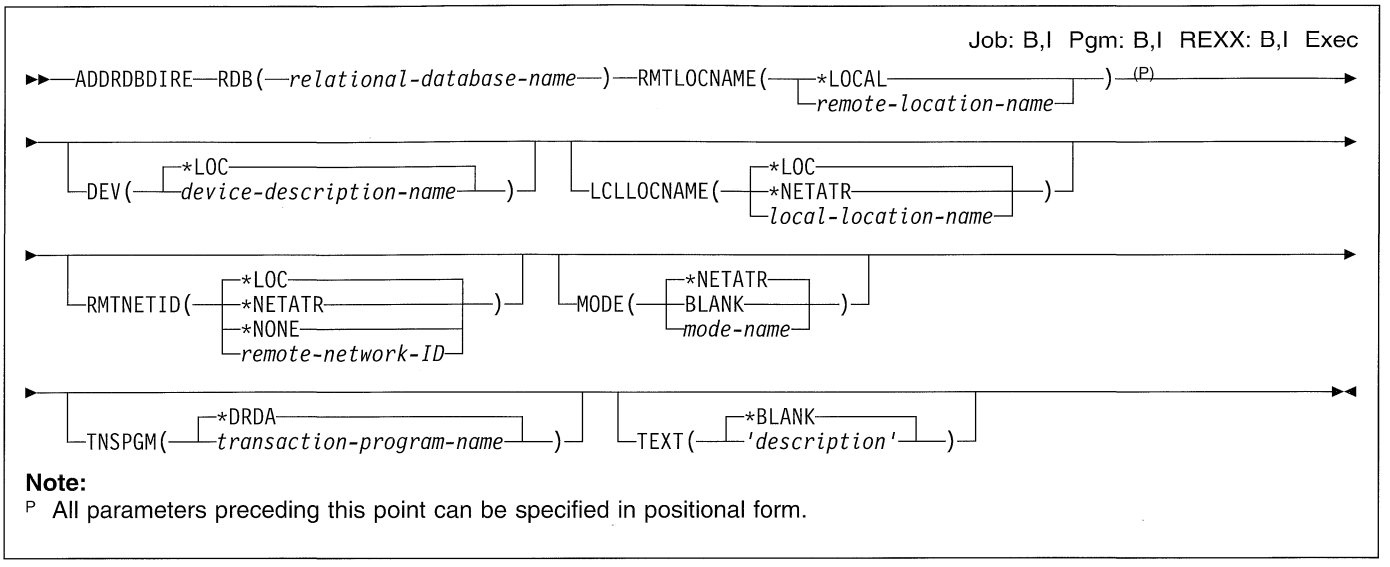
   ```
   ((*IF  *PROBTYPE *EQ 1) +
    (*AND *SEV *EQ 1)     +
    (*OR  *PROBTYPE *EQ 2) +
    (*AND *SEV *EQ 1))
   ```

   is equivalent to the following Boolean expression:

   ```
   if ((*PROBTYPE = 1) and (*SEV = 1)) or
      ((*PROBTYPE = 2) and (*SEV = 1))
   ```

5. All attribute values are interpreted as character data, including numbers. When the problem filter is applied to a problem, the system converts all of the data in the filter to the type given in the problem template and compared. Message ID's are considered character data and are ordered as such.

ADDRDBDIRE (Add Relational Database Directory Entry) Command



Note:

| P All parameters preceding this point can be specified in positional form.

Purpose

The Add Relational Database Directory Entry (ADDRDBDIRE) command adds a relational database entry to the relational database directory.

Each entry identifies the method of accessing the relational database. One entry naming the local relational database and one entry for each remote relational database must be added. Each relational database name must be unique within the relational database directory and within the distributed network.

Required Parameters

RDB

Specifies the name of the relational database being added. Each name entered into the directory must be unique. Because relational databases communicate with each other over a distributed network, each name must also be unique among other relational databases on the network. The system can determine whether a relational database name is unique in the directory, but you must determine whether it is unique among relational databases on the network. A maximum of 18 characters can be specified for the relational database name. If an MVS(DB2*) relational database is identified, only 16 characters are allowed.

Note: Valid RDB names must begin with a letter and consist of uppercase A-Z, 0-9, and underscore.

RMTLOCNAME

Specifies the remote location name of the system on which the relational database is located.

***LOCAL:** The relational database is located on the local system. *LOCAL can be specified for only one entry in the relational database directory since there is only one relational database on the local system.

remote-location-name: Specify a maximum of 8 characters for the remote location name. More information about remote location names can be found in the *APPC Programmer's Guide* and the *APPN Guide*.

Optional Parameters

DEV

Specifies the name of the advanced program-to-program communications (APPC) device description on the local system that is used to access this relational database. The device description does not need to exist when the relational database directory entry is added.

More information on device names is in the *APPC Programmer's Guide*.

***LOC:** If APPC is being used, the system determines which device description is used. If advanced peer-to-peer networking (APPN) is being used, the system ignores this parameter.

device-description-name: Specify a maximum of 10 characters for the name of a device description.

LCLLOCNAME

Specifies the local location name by which the local system is identified to the system on which the relational database is located. The local location name cannot be the same as the remote location name.

More information on local location names is in the *APPC Programmer's Guide*.

***LOC:** If APPC is being used, the system determines which local location name is used. If APPN is being used, the system uses the default local location name specified in the network attributes.

***NETATR:** The local location name defined in the network attributes is used.

local-location-name: Specify a maximum of 8 characters for the local location name.

RMTNETID

Specifies the remote network identifier of the system on which the relational database is located.

More information on remote network identifiers is in the *APPC Programmer's Guide*.

***LOC:** If APPC is being used, the system determines which remote network identifier is used. If APPN is being used, the system uses the local network identifier defined in the system's network attributes for the remote network identifier.

***NETATR:** The local network identifier defined in the local system's network attributes is used for the remote network identifier.

***NONE:** No remote network identifier is used.

remote-network-identifier: Specify a maximum of 8 characters for the remote network identifier.

MODE

Specifies the mode name that is used with the remote location name to communicate with the system on which the relational database is located.

***NETATR:** The mode name defined in the network attributes is used.

***BLANK:** A mode name of all blanks is used.

mode-name: Specify a maximum of 8 characters for the mode name.

More information on mode names is in the *APPC Programmer's Guide*.

TNSPGM

Specifies the name of the transaction program to use with the relational database entry.

***DRDA:** The Distributed Relational Database Architecture* (DRDA*) transaction program name, X'07F6C4C2', is used. DRDA is a means by which relational databases communicate with each other over a distributed network.

transaction-program-name: Specify the transaction program name in either of the following formats:

- A 4-byte hexadecimal name, which is entered by enclosing the 8 hexadecimal digits in apostrophes with a prefix of X. For example, X'07F6C4C2' is a 4-byte hexadecimal name.
- An 8-byte character name, which is entered by specifying the name in its 8-character form.

TEXT

Specifies text that briefly describes the relational database. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***BLANK:** No text is associated with the new relational database.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Examples

Example 1: Adding an Entry

```
ADDRDBDIRE   RDB(MYRDB)  RMTLOCNAME(*LOCAL)
```

This command adds an entry to the relational database directory. The entry identifies the local relational database. In an SQL program, this relational database name is used when referring to the local relational database.

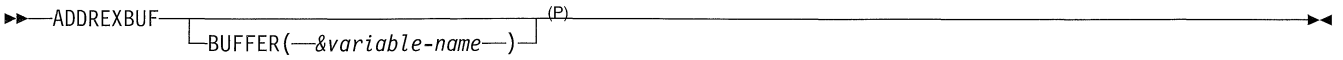
Example 2: Adding an Entry

```
ADDRDBDIRE   RDB(YOURRDB)  RMTLOCNAME(NEWYORK)
```

This command adds an entry to the relational database directory. The entry identifies a remote location, NEW YORK.

ADDREXBUF (Add REXX Buffer) Command

Pgm: B,I REXX: B,I



Note:

P All parameters preceding this point can be specified in positional form.

Purpose

The Add REXX Buffer (ADDREXBUF) command allows the user to create a buffer in the REXX external data queue.

Optional Parameter

BUFFER

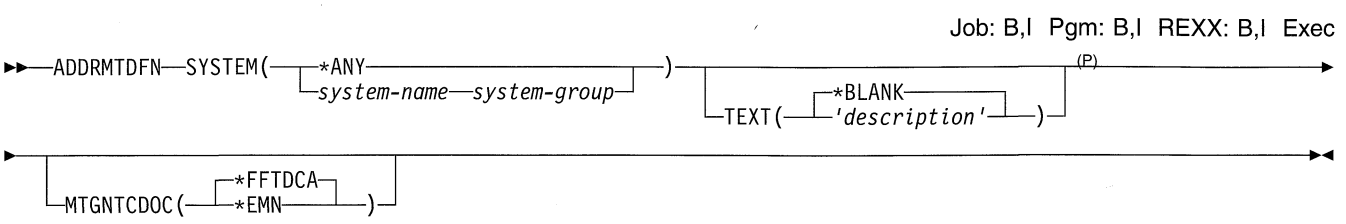
Specifies the name of the variable that receives the number of the new buffer. In a control language (CL) program, a decimal variable with a minimum length of 11 digits and no decimal position must be specified.

Example

```
ADDREXBUF
```

This command creates a logical buffer within the REXX external data queue.

ADDRMTDFN (Add Remote Definition) Command



Note:

P All parameters preceding this point can be specified in positional form.

Purpose

The Add Remote Definition (ADDRMTDFN) command is used to define the attributes of a remote system.

Restriction: The user must have *ALLOBJ authority.

Required Parameter

SYSTEM

Specifies the system name and system group of the remote system being defined.

***ANY:** Adds a default definition for all remote systems not covered by the other entries.

Element 1: System Name

system-name: Specify the name of the remote system being defined.

Element 2: System Group

system-group: Specify the group name of the remote system being defined. The system group name is blank if this value is not specified.

Optional Parameters

TEXT

Specifies text that briefly describes the remote system definition. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

MTGNTCDOC

Specifies the type of meeting notice documents accepted by the remote system. If the system can accept Enterprise Meeting Notice Architecture documents (post-V2R1.1 AS/400 systems), you should specify *EMN for this parameter. If you are unsure, specify *FFTDCA for this parameter.

***FFTDCA:** The remote system does not accept enterprise meeting notice documents, but does accept final-form text documents.

***EMN:** The remote system accepts enterprise meeting notice documents.

Examples

Example 1: Adding a Specific Remote Definition

```
ADDRMTDFN  SYSTEM(ABCXYZ)  TEXT('System XYZ')
  MTGNTCDOC(*EMN)
```

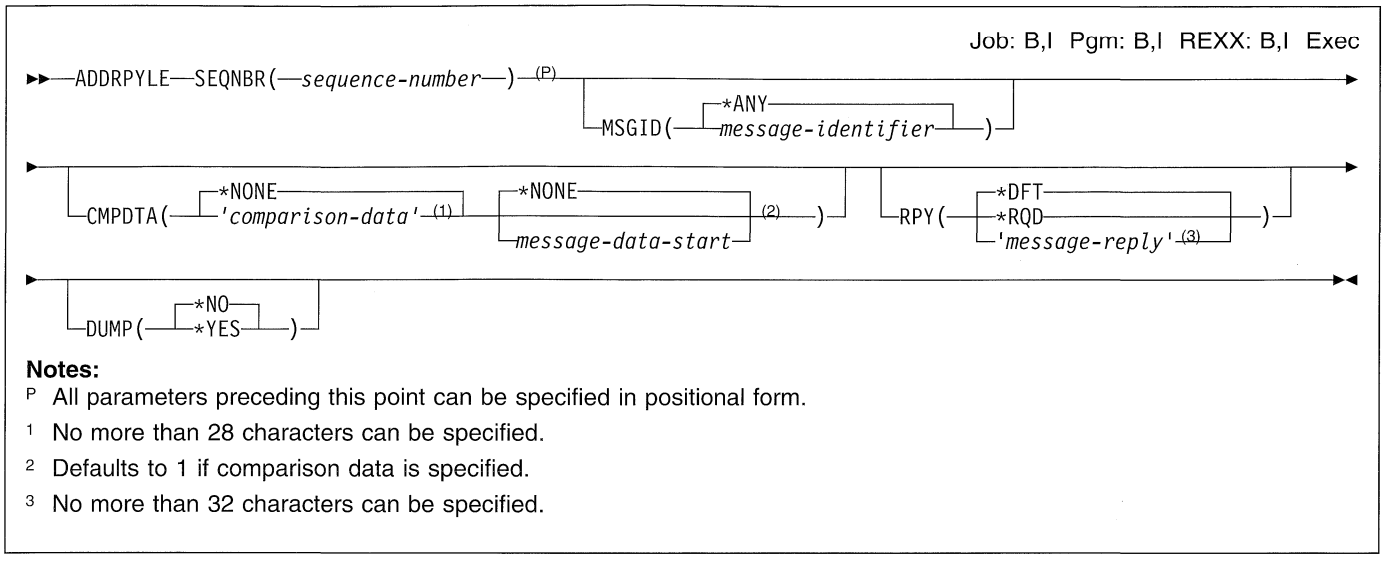
This command adds a definition for a remote system ABCXYZ and allows the system to accept Enterprise Meeting Notice documents.

Example 2: Allowing Final Form Text Documents

```
ADDRMTDFN  SYSTEM(*ANY)  MTGNTCDOC(*FFTDCA)
```

This command allows all remote systems that do not have specific remote definitions to accept final form text meeting notices.

ADDRPYLE (Add Reply List Entry) Command



Notes:

P All parameters preceding this point can be specified in positional form.

1 No more than 28 characters can be specified.

2 Defaults to 1 if comparison data is specified.

3 No more than 32 characters can be specified.

Purpose

The Add Reply List Entry (ADDRPYLE) command is used to add an entry to the system-wide automatic inquiry message reply list. The automatic message reply list is the source for default responses to inquiry messages. Each entry in the inquiry message list specifies both a message identifier and the reply that is sent when that message is sent as an inquiry message. The entry may also include comparison data, which further qualifies the message identifier. The message identifier may be specific or generic in scope. One of the following actions may be taken when one of the specific inquiry messages is issued:

- The default reply specified in the inquiry message file is sent to the message reply queue specified when the inquiry message was sent.
- A specific reply to the inquiry message is sent to the message reply queue specified when the inquiry message was sent.
- A manual reply to the inquiry message may be required from the operator.

The entry may also specify the dumping of information associated with the job that is sending the inquiry message.

The reply list is used only when an inquiry message is sent by a job that has the inquiry message reply attribute specified as INQMSGRPY(*SYSRPYL). The INQMSGRPY attribute can be changed by using the CHGJOBD command.

Specific attributes of a reply list entry can be changed by using the Change Reply List Entry (CHGRPYLE) command. Each reply list entry remains in the list until it is removed by the Remove Reply List Entry (RMVRPYLE) command. The list can be shown by using the Work with Reply List Entry (WRKRPYLE) command.

Restriction: This command is shipped with public *EXCLUDE authority and the QPGMR user profile has private authority to use the command.

Required Parameter

SEQNBR

Specifies the sequence number of the reply list entry being added to the reply list. The message identifier and message data of an inquiry message are matched against the reply list entry message identifiers and comparison data in ascending sequence number order. The search ends when a match occurs or when the last reply list entry is passed. Therefore, if more than one reply list entry matches the inquiry message identifier and comparison data, only the first entry that matches is used. If no reply list entry matches the inquiry message, the inquiry is sent, but no automatic reply is sent unless a default reply is sent, and the information associated with the job is not dumped. Default replies are sent when the delivery mode of the message queue is set to *DFT or when the inquiry message is sent to an external message queue in a batch job.

Sequence numbers range from 0001 through 9999 and duplicate sequence numbers are not allowed.

Optional Parameters

MSGID

Specifies the inquiry message identifiers for which automatic system action is taken. The message identifier can be either specific or generic in scope. Only predefined messages (messages known to the system by a message identifier) can be matched by reply list entries. Impromptu messages cannot be used for comparison.

If no comparison data is specified, then only the message identifier is used to match the message to this

reply list entry. If this is the first message reply list entry that matches the message, the action specified in this entry is taken.

***ANY:** This reply list entry matches all message identifiers. Unless comparison data is specified for this reply list entry, all reply list entries with a sequence number greater than this one are ignored.

message-identifier: Specify a reply message identifier that is compared with the inquiry message identifier. The message identifier must be seven characters in length and in the format, pppnnnn.

The first three characters (ppp) must be a code consisting of one alphabetic character followed by two alphanumeric (alphabetic or decimal) characters. The last four characters (nnnn) may consist of the decimal numbers 0 through 9 and the characters A through F.

To specify a generic message identifier, enter zeros in the rightmost two or four positions of the numeric field, such as pppnn00 or ppp0000. For example, CPA0000 would match any CPA inquiry message, while CPA4200 would match any CPA42xx inquiry message.

CMPDTA

Specifies the comparison data that is used to determine whether this entry matches an inquiry message. This parameter is made up of comparison data and a message data start value. If the identifier of the inquiry message matches the message identifier of this reply list entry, then the message data specified for the inquiry message is compared to this data. If a message data start value has not been specified, then the first part of the message data (up through the first 28 characters or less) must exactly match the comparison data specified here before the action requested for this reply list entry is taken. However, if a start value has been specified, then the part of the message data beginning with the character position specified in the start value must exactly match the comparison data before any requested action is taken. If the comparison data is longer than the message data, then no match occurs. If no comparison data is specified, then only the message identifier is used to match the message to this reply list entry. If, in the message reply list, this is the first entry that matches the message, then the action specified in this entry is taken.

Message data for an inquiry message may be specified in the MSGDTA parameter of the SNDUSRMSG or SNDPGMMSG commands for the inquiry message.

Element 1: Comparison Data

***NONE:** No comparison data is specified; if the inquiry message has the specified identifier, the action specified by this reply list entry is taken.

'comparison-data': Specify a character string of up to 28 characters (enclosed in apostrophes if blanks or other special characters are included). This string is compared with a string of the same length in the message data portion of the inquiry message, beginning with the first character (if no start value has been specified). If the comparison data string matches the inquiry message data string, the action specified by this reply list entry is taken.

Element 2: Message Data Start Position

***NONE:** No message data start value is specified. If a comparison value is specified and a message data start value is not specified, a default value of 1 is used.

message-data-start: Specify the character position in the message's replacement text (maximum value is 999) where the comparison of the comparison data and the replacement text starts. A start value is not valid without a specification of comparison data.

RPY

Specifies how to reply to an inquiry message that matches this reply list entry. The reply specified (other than *RQD) in this reply list entry is automatically sent by the system without requiring user intervention. The inquiry message does not cause the job to be interrupted or notified when the message arrives at the message queue. The inquiry message is not shown before the reply message is sent and there is no opportunity for a manual reply to the inquiry message.

***DFT:** The default reply to the inquiry message is sent. If no default reply is specified in the message description of the inquiry message, the system default reply, *N, is used.

***RQD:** The inquiry message requires an explicit reply. If the message queue to which the inquiry is sent is in break mode, the message interrupts and is displayed. If the message queue is in the notify mode, the job to which it is allocated is notified. No reply is automatically sent. If the message queue is in the hold mode, the message remains on the queue until it is removed.

'message-reply': Specify a character string of up to 32 characters, enclosed in apostrophes if blanks or other special characters are included, that is sent as a reply to the inquiry message. If this reply is not valid for the inquiry message, the inquiry is sent as if RPY(*RQD) had been specified.

DUMP

Specifies whether the job that sent the inquiry message is dumped. The dump is the same as the dump specified by DMPLST(*JOB) on the Add Message Description (ADDMSGD) command or by OUTPUT(*PRINT) on the Display Job (DSPJOB) command for the sending job. A job dump may be requested regardless of the value specified for the RPY parameter.

***NO:** The job is not dumped.

***YES:** The job is dumped before control returns to the program that is sending the message.

Examples

Example 1: Reply Automatically Sent

```
ADDRPYLE   SEQNBR(10)  MSGID(RPG1241)  RPY(G)
```

This command adds a reply list entry to the reply list for message identifier RPG1241 (database record not found). Whenever a RPG1241 inquiry message is sent by a job that is using the reply list, a reply of 'G' is automatically sent. The inquiry does not cause a job that has allocated the message queue to be interrupted or notified when the inquiry arrives, and no opportunity is given to reply to the message. The sending job does not have a job dump processed.

Example 2: Default Reply is sent; Job Dump Processed

```
ADDRPYLE   SEQNBR(25)  MSGID(RPG1200)
  RPY(*DFT)  DUMP(*YES)
```

This command adds a generic reply list entry to the reply list for all RPG12xx messages. Whenever an RPG12xx inquiry message is sent by a job that is using the reply list, the equivalent to DSPJOB OUTPUT (*PRINT) is automatically generated. The default reply will automatically be sent. This is either the default reply specified in the message description or (if none is specified in the message description) the system default reply. The inquiry does not cause a job that has allocated the message queue to be interrupted or notified when the inquiry arrives, and no opportunity is given to reply to the message. The sending job is dumped before control returns to the sending program. Note that because of the sequence numbers, the entry added by the previous example overrides this entry for message identifier RPG1241.

Example 3: Adding a Generic Reply List Entry

```
ADDRPYLE   SEQNBR(30)  MSGID(RPG0000)
  RPY(D)  DUMP(*YES)
```

This command adds a generic reply list entry to the reply list for all RPG messages. Whenever an RPG inquiry message is sent by a job that is using the reply list, a reply of 'D' is sent automatically. The inquiry does not cause a job that has allocated the message queue to be interrupted or notified when the inquiry arrives, and no opportunity is given to reply to the message. (If a value of D is not valid for a particular RPGxxxx message, the user must reply as if *RQD were specified for the RPY parameter.) The sending job is dumped before control returns to the sending program. Note that the entries added by the previous two examples will override this entry for all RPG12xx messages.

Example 4: System Reply List for Spooled Output

```
ADDRPYLE   SEQNBR(40)  MSGID(CPA5316)
  CMPDTA('QPSPLPRT  QSYS      QSYSPRT')  RPY(*RQD)
```

This command illustrates how to use the system reply list for spooled output for device QSYSPRT. The file and library name for spooled output is QSYS/QPSPLPRT.

When compare value is specified, it is compared to the message data beginning with replacement variable &1. If the significant field appears in replacement variable &3, the compare value must include a value for replacement variables &1 and &2, or a message data start value may be entered to begin the comparison with replacement variable &3.

The message CPA5316 has a replacement data as follows:

```
| &1        ODP file name        &CHAR  10
| &2        ODP library name     &CHAR  10
| &3        ODP device name      &CHAR  10
```

A compare for device name 'QSYSPRT' in replacement variable &3 must be preceded by values for &1 and &2 if a message data start value is not entered. Blanks are significant.

The message data of QSYSPRT is the DEVICE name as defined in the CPA5316 message. Whenever a CPA5316 inquiry message with comparison data of QSYSPRT is sent by a job that is using the reply list, the operator must make a manual reply to the inquiry. If the message queue to which the inquiry is sent is in break mode, the inquiry message interrupts. A reply is not sent (unless the queue is in the default mode or the message is sent to an external message queue in a batch job), and no job dump is taken.

Another reply list entry identical to the one listed above could be added, but with a different sequence number and with CMPDTA(WSPR01) specified. This would allow a unique response to a message based on the type of printer.

Example 5: Adding Reply List Entry For Any Message Identifier

```
ADDRPYLE   SEQNBR(9999)  MSGID(*ANY)
  RPY(*RQD)  DUMP(*YES)
```

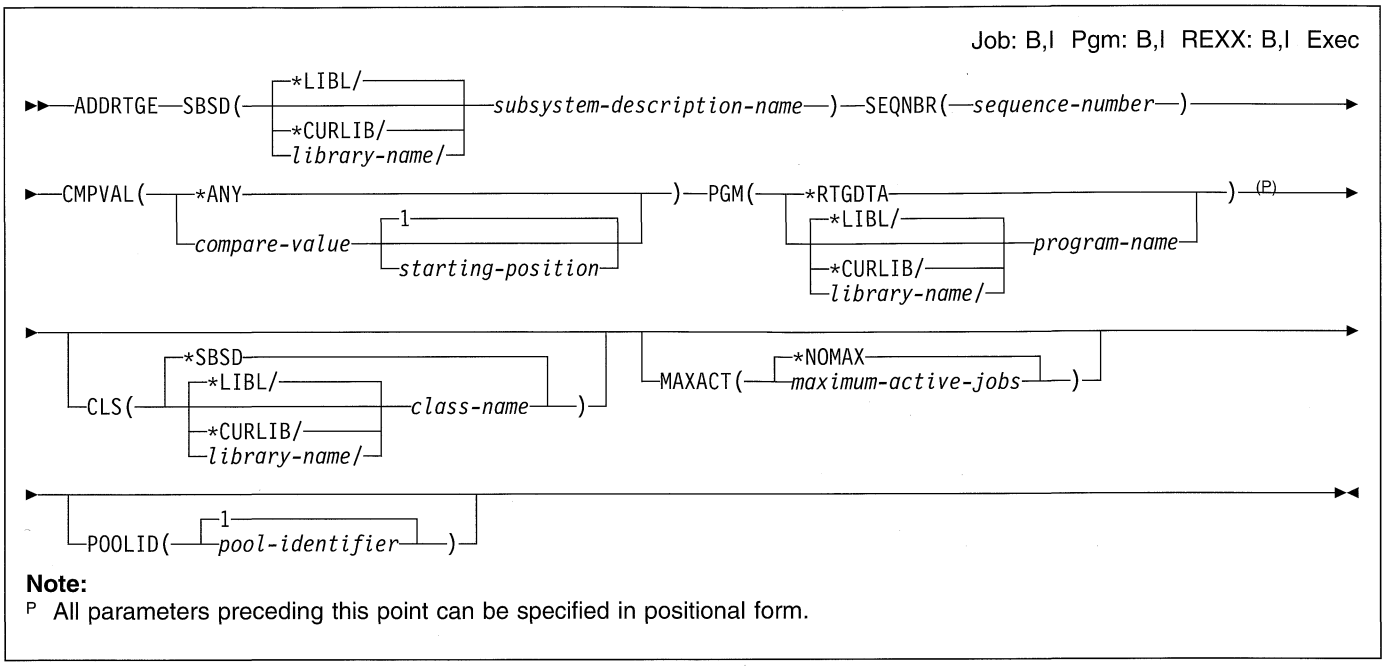
This command adds a reply list entry to the reply list for any message identifier. This entry applies to any predefined inquiry message that is not matched by an entry with a lower sequence number. A manual reply to the inquiry message is required for any predefined inquiry message not matched by a previous entry. If the message queue to which the inquiry message is sent is in break mode, the message interrupts. The job that sent the inquiry message is dumped (equivalent to DSPJOB OUTPUT(*PRINT)).

Example 6: Using Comparison Data

```
ADDRPYLE SEQNBR(5) MSGID(CPA5316) CMPDTA(QSYSPRT 21)
  RPY(I) DUMP(*NO)
```

Assume that the message CPA5316 is sent to QSYSOPR with the message replacement text of TESTEDFILETSTLIBRARYQSYSPRT; because there is a match for MSGID, the message replacement text starting in position 21 (message data start) is tested by comparing it with the comparison data (for the length of the comparison data). This is a match because QSYSPRT = QSYSPRT, and therefore the reply of 'I' is sent.

ADDRTGE (Add Routing Entry) Command



Note:

| P All parameters preceding this point can be specified in positional form.

Purpose

The Add Routing Entry (ADDRTGE) command adds a routing entry to the specified subsystem description. The associated subsystem must be inactive at the time. Each routing entry specifies the parameters used to start a routing step. For example, the routing entry specifies the name of the program to run when the routing data that matches the compare value in this routing entry is received.

Restriction: To use this command, the user must have object operational and object management authorities for the subsystem description.

Required Parameters

SBSD

Specifies the qualified name and library of the subsystem description to which the routing entry is added.

| The name of the subsystem description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

subsystem-description-name: Specify the name of the subsystem description to which the routing entry is added.

SEQNBR

Specifies the sequence number of the routing entry that is added. Routing data is matched against the routing entry compare values in ascending sequence number order. Searching ends when a match occurs or the last routing entry is compared. Therefore, if more than one match possibility exists, only the first match is processed. Specify a unique sequence number (ranging from 1 through 9999) that identifies the routing entry.

CMPVAL

Specifies a value that is compared with the routing data to determine whether this is the routing entry used for starting a routing step for the job. If the routing data matches the routing entry compare value, that routing entry is used. Optionally, a starting position in the routing data character string can be specified for the comparison.

Element 1: Comparing Values with Routing Data

***ANY:** Any routing data is considered a match. To specify *ANY, the routing entry must have the highest SEQNBR value of any routing entry in the subsystem description.

compare-value: Specify a value (any character string not exceeding 80 characters in length) that is compared with routing data for a match. When a match occurs, this routing entry is used to start a routing step. A starting position in the routing data character string can be specified for the comparison; if no position is specified, 1 is assumed.

Element 2: Starting Position

1: The comparison between the compare value and the routing data begins with the first position in the routing data character string.

starting-position: Specify a value, ranging from 1 through 80, that indicates which position in the routing data character string is the starting position for the comparison. The last character position compared must be less than or equal to the length of the routing data used in the comparison.

PGM

Specifies the qualified name of the program called as the (first) program run in the routing step. No parameters can be passed to the specified program.

The program name can be either explicitly specified in the routing entry or extracted from the routing data. If a program name is specified in a routing entry, selection of that routing entry results in the routing entry program being called regardless of the program name passed in an EVOKE function. If the program specified in the EVOKE function is called, PGM(*RTGDTA) must be specified in the routing entry. If the program does not exist when the routing entry is added, a library qualifier must be specified because the qualified program name is retained in the subsystem description.

***RTGDTA:** The program name is taken from the routing data that was supplied and matched against this entry. A qualified program name is taken from the routing data in the following manner: the program name is taken from positions 37 through 46, and the library name is taken from positions 47 through 56. Care should be used to ensure that routing entries that specify *RTGDTA are selected only for EVOKE functions on jobs that have specified the program name in the correct position in the routing data.

The name of the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the program that is run from this routing entry.

Optional Parameters

CLS

Specifies the qualified name of the class used for the routing steps started through this routing entry. The class defines the attributes of the running environment for processing the routing step associated with this routing entry. If the class does not exist when this routing entry is changed, a library qualifier must be specified because the qualified class name is retained in the subsystem description. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SBSD:** The class having the same qualified name as the subsystem description specified by the SBSD parameter is used for routing steps started through this entry.

The name of the class can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

class-name: Specify the name of the class that is used for routing steps started through this routing entry.

MAXACT

Specifies the maximum number of routing steps (jobs) that can be active at the same time through this routing entry. In a job, only one routing step is active at a time. When a subsystem is active and the maximum number of routing steps is reached, any subsequent attempts to start a routing step through this routing entry fails. If the routing data is entered interactively, an error message is sent to the user. Otherwise, the job is ended, and a message is sent by the subsystem to the job's log. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***NOMAX:** There is no maximum number of routing steps that can be active at the same time and be processed through this routing entry. This value is normally used when there is no reason to control the number of routing steps.

maximum-active-jobs: Specify the maximum number of routing steps that can be active at the same time and be processed through this routing entry. If a routing step being started would exceed this number, the job is implicitly ended.

POOLID

Specifies the pool identifier of the storage pool in which the program runs.

1: Storage pool 1 of this subsystem is the pool in which the program runs.

pool-identifier: Specify the identifier of the storage pool defined for this subsystem in which the program runs. Valid values range from 1 through 10.

Examples

Example 1: Adding to the Routing Portion of a Subsystem Description

```
ADDRTGE  SBSD(ORDLIB/PERT)  SEQNBR(46)
  CMPVAL(WRKSTN2)  PGM(ORDLIB/GRAPHIT)
  CLS(MYLIB/AZERO)  MAXACT(*NOMAX)  POOLID(2)
```

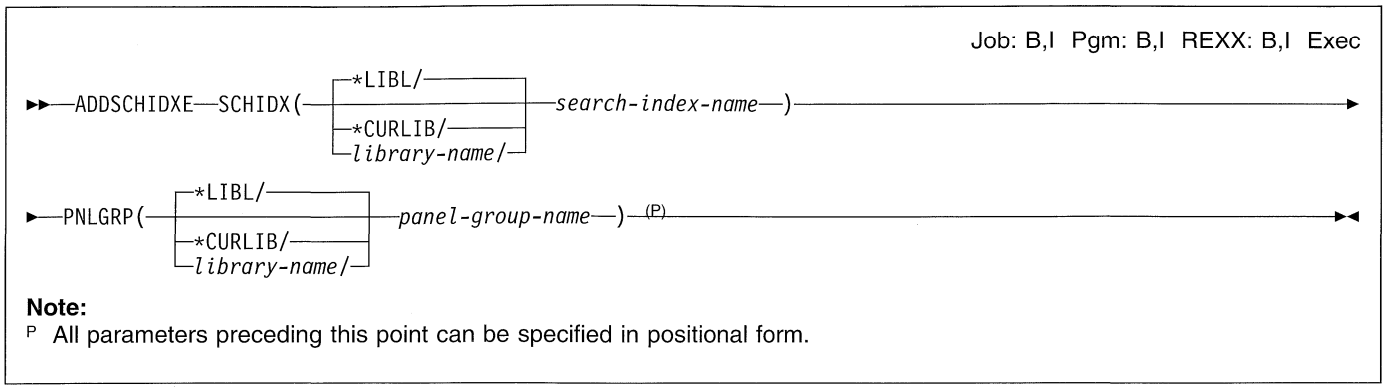
This command adds routing entry 46 to the routing portion of subsystem description PERT in the ORDLIB library. To use routing entry 46, the routing data must start with the character string WRKSTN2 starting in position 1. Any number of routing steps can be active through this entry at any one time. The program GRAPHIT in the library ORDLIB is to run in storage pool 2 by using class AZERO in library MYLIB.

Example 2: Adding to the Subsystem Description

```
ADDRTGE  SBSD(QGPL/ABLE)  SEQNBR(5)  CMPVAL(XYZ)
  PGM(QGPL/REORD)  CLS(LIBX/MYCLASS)  MAXACT(*NOMAX)
```

This command adds routing entry 5 to the subsystem description ABLE in the QGPL library. The program REORD in the general purpose library is started and uses the class MYCLASS in LIBX when a compare value of XYZ (starting in position 1) is matched in the routing data. The program runs in storage pool 1, and there is no maximum on the number of active routing steps allowed.

ADDSCHIDXE (Add Search Index Entry) Command



Note:
(P) All parameters preceding this point can be specified in positional form.

Purpose

The Add Search Index Entry (ADDSCHIDXE) command is used to load help text topics into a search index.

A search index refers to help text from one or more panel groups. A panel group contains help text, which the user can access from display panels by pressing the Help key, or through the information search function using the Start Search Index (STRSCHIDX) command.

The sequence in which panel groups are loaded into a search index controls the sequence in which topic entries are presented when an information search is requested. The topics (ISCH tag entries) from the first-loaded panel group are presented first.

Restrictions: (1) The user must have *CHANGE authority for the search index that is being loaded and *USE authority for the panel group. (2) Only user-created panel groups can be added to user-created search indexes and only IBM-supplied panel groups can be added to IBM-supplied search indexes. (3) Panel group names must be unique within a search index.

Required Parameters

SCHIDX

Specifies the qualified name of the search index into which entries are loaded.

The name of the search index can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

search-index-name: Specify the name of the search index.

PNLGRP

Specifies the qualified name of the panel group that contains the help text for which entries are to be loaded into the search index.

The search index object contains the name and library of the panel group. When help text is displayed for a search index, the name and library of the panel group that is contained in the search index object is used to find the panel group.

When *LIBL is used to qualify the panel group name, *LIBL is saved in the search index object. When the panel group name is qualified with either a library name or *CURLIB, the name of the library containing the panel group is saved in the search index object.

The names of panel groups added to the search index must be unique.

The name of the panel group can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

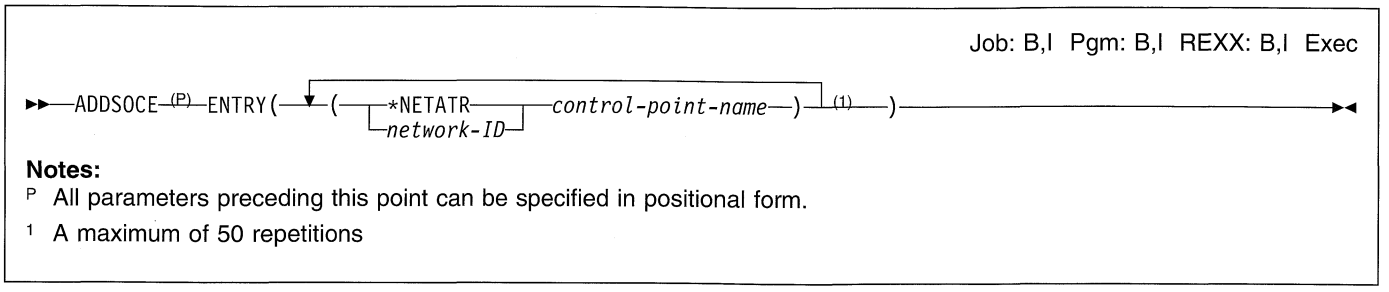
panel-group-name: Specify the name of the panel group.

Example

```
ADDSCHIDXE   SCHIDX(ACCOUNTING)  PNLGRP(PAYROLL)
```

This command adds panel group PAYROLL to search index ACCOUNTING. Both the panel group and the search index must exist in the library list.

ADDSOCE (Add Sphere of Control Entry) Command



Purpose

The Add Sphere of Control Entry (ADDSOCE) command allows a CL user or program to add control points to the Alert Sphere of Control.

Required Parameter

ENTRY

Specifies the systems to add to the sphere of control. The systems are specified as a list of two elements that includes the network ID and control point name.

Element 1: Network ID of the System

***NETATR:** The NETID network attribute is used as the value of the network ID being added to the alert sphere of control.

network-ID: Specify the network ID of system being added to the alert sphere of control.

Element 2: Control Point Name of the System

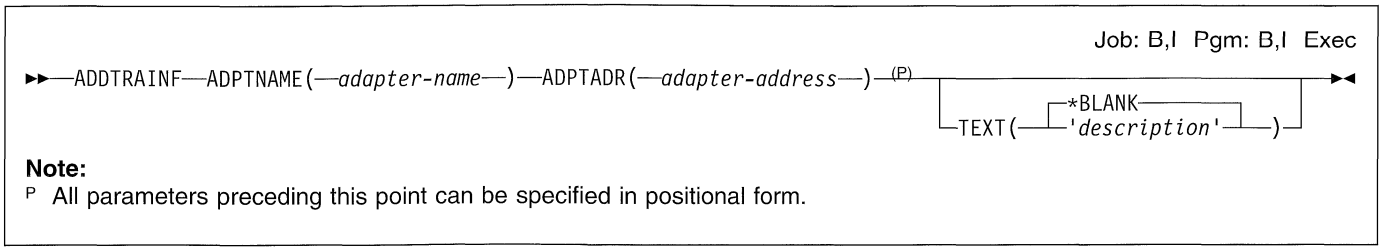
control-point-name: Specify the control point name of the system being added to the alert sphere of control.

Example

```
ADDSOCE  ENTRY((*NETATR RCHSTR1) (*NETATR RCHSTR2))
```

This command adds two systems (RCHSTR1 and RCHSTR2) to the alert sphere of control.

ADDTRAINF (Add TRLAN Adapter Information) Command



Note:

| P All parameters preceding this point can be specified in positional form.

Purpose

The Add Token-Ring Local Area Network Adapter Information (ADDTRAINF) command adds an adapter name entry to the adapter file.

Required Parameters

ADPTNAME

Specifies the name of the adapter being added to the adapter file.

ADPTADR

Specifies the 12-character hexadecimal adapter address.

Optional Parameter

TEXT

Specifies text that briefly describes the adapter. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***BLANK:** Text is not specified.

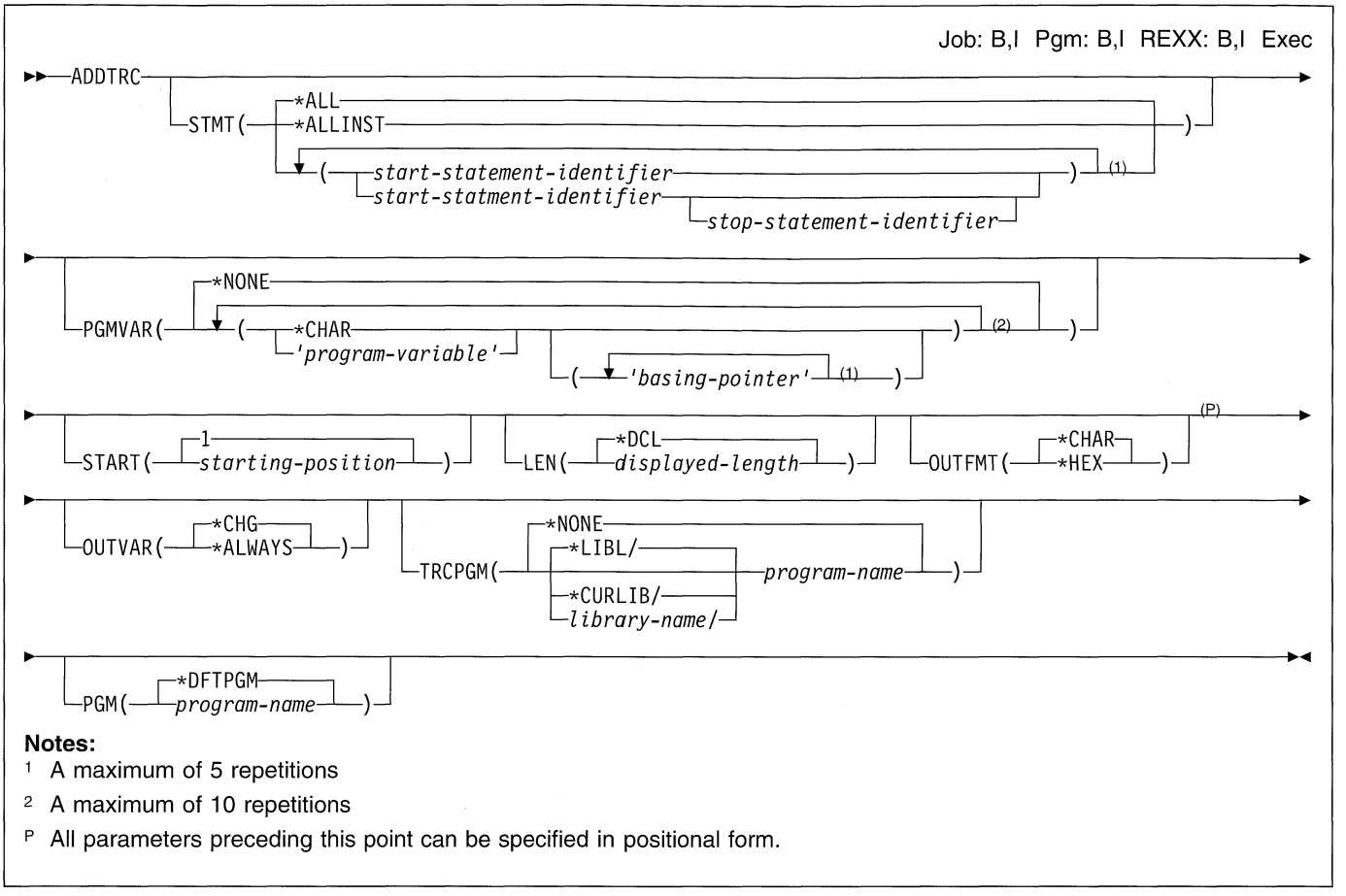
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
ADDTRAINF  ADPTNAME(PAYROLL)  ADPTADR(00000000012B)
```

This command adds adapter PAYROLL with address 00000000012B to the adapter file.

ADDTRC (Add Trace) Command



Notes:

1 A maximum of 5 repetitions

2 A maximum of 10 repetitions

P All parameters preceding this point can be specified in positional form.

Purpose

The Add Trace (ADDTRC) command specifies which program statements in a program to trace in debug mode. Up to five ranges of high-level language (HLL) statements or machine instructions can be traced during the processing of a program through one or more ADDTRC commands, and up to 10 program variables can be recorded or monitored for change in each specified statement range. A separate ADDTRC command is required for each unique variable associated with a statement range. When the specified program being traced is run, the system records the sequence in which the traced statements are processed and optionally records the value of the variables associated with the trace each time a traced statement is processed. After a trace has been completed, the user can display this information using the Display Trace Data (DSPTRCDTA) command.

All of the trace ranges specified in a program are active at the same time. If both a HLL statement identifier and a machine instruction number are used to specify a given trace range, the trace range is treated as an HLL trace range. That is, in addition to tracing the machine instruction number specified, the system traces the HLL statement identifiers between that machine instruction number and the specified HLL statement identifier. More information on testing and debugging at the machine interface level is in the *CL Programmer's Guide*.

Restrictions:

1. This command is valid only in debug mode. To start debug mode, refer to the STRDBG (Start Debug) command.
2. This command cannot be used if the user is servicing another job, and that job is on a job queue, or is being held, suspended, or ended.
3. This command cannot be used to trace bound programs.

Optional Parameters

STMT

Specifies which program statements (or machine instructions) to trace in one or more statement ranges in the program being traced.

***ALL:** All statements in the specified high-level language program are traced.

***ALLINST:** All machine instructions in the specified program are traced.

start-statement-identifier [stop-statement-identifier]: Specify the HLL statement identifiers (or machine

instruction numbers) at which tracing starts and, optionally, the identifier at which tracing stops. Up to five trace ranges can be defined at the same time for any program in debug mode. Each trace range begins with the specified starting statement, and all following statements are traced until the ending statement is reached. If only a starting statement identifier is specified for a range, the single statement specified is the only statement traced for that range. If machine instruction numbers are specified, a slash must be placed in front of the number and both the slash and the number must be enclosed in apostrophes; for example:

```
STMT(('/21' '/43')('/62' '/98'))
```

In high-level language programs, different statements and/or labels can be mapped to the same internal instruction. This happens when there are several statements that do not operate on variables directly (such as DO, END, or comments) following one another in a program. To determine which statements (labels) can be mapped to the same instruction, the intermediate representation of a program list can be used.

PGMVAR

Specifies the names of the variables whose values are recorded when a trace statement in a program is processed. The OUTVAR parameter determines whether the values can be recorded for every trace statement processed or only when a variable changes value. The program variables can be specified either by their HLL names or by their machine-interface object-definition-table-vector (MI ODV) numbers.

Element 1: Program Variables

***NONE:** No program variables have their values recorded during tracing.

***CHAR:** This value is used instead of a variable name if a basing pointer is also specified. This special value displays a character view of a pointer to be shown without the use of a based variable.

'program-variable': Specify the names of one to ten program variables whose values are recorded during tracing. If a variable name contains special characters, it must be enclosed in apostrophes. For example, a CL variable, &VAR, must be specified as `PGMVAR('&VAR')`.

If the program variable is an array, the subscripts representing an element in the array can be specified. If an array name is specified without any subscripts, all of the array elements are recorded. A single-dimensional cross-section can also be specified. Up to 132 characters may be specified for this program variable entry. This includes any qualifiers, subscripts, embedded blanks, parentheses, and commas. It does not include the enclosing apostrophes when special characters are used. An integer, MI ODV number, asterisk (single-dimensional cross-section), or a numeric variable name can be specified for a subscript. For more information on the program-variable value, refer to Appendix C, "Parameter Values Used for Testing and Debugging."

Some examples are:

```
PGMVAR(A)
PGMVAR('A(2,B)')
PGMVAR('B(I1,*,I3)')
PGMVAR('VAR1 OF A(I,J IN B)')
```

Element 2: Basing Pointers

'basing-pointer': Specify up to five basing pointers for the program variable being displayed. In some languages, the program variable may be based on a pointer variable. This set of values allows the user to specify the basing-pointers for the variable to be recorded. Each basing-pointer name must be enclosed in apostrophes if it contains special characters.

If the basing-pointer is an array, the subscripts representing an element in the array must be specified. Up to 132 characters can be specified for a basing-pointer name. This includes any qualifiers, subscripts, embedded blanks, parentheses, and commas. It does not include the enclosing apostrophes when special characters are used. An integer, MI ODV number, or a numeric variable name can be specified for a subscript. For more information on the basing-pointer value, refer to Appendix C, "Parameter Values Used for Testing and Debugging." Some examples are:

```
PGMVAR(('VAR1(B,5)' 'PTR2(C,P2)'))
PGMVAR((VAR2 (BASEPTRA BASEPTRB)))
```

START

Specifies, for string variables only, the starting position in the string from which its value is recorded during tracing. If more than one string variable is specified in the PGMVAR parameter, the same starting position value is used for each one. For a bit string, the value specifies the starting bit position; for a character string, the value specifies the starting character position.

1: The variable is shown from the first position through the length specified on the LEN parameter.

starting-position: Specify the first position being recorded in the program variable.

The START value specified must not be larger than the maximum string length for any variable specified, except that START(1) is allowed if the maximum length for a string is zero. The LEN value, plus the START position minus one, must not be greater than the maximum string length. These checks are made for each string variable specified in the PGMVAR parameter.

LEN

Specifies, for string variables only, the length of the string being recorded during the trace, starting at the position specified in the START parameter. If more than one string variable is specified in the PGMVAR parameter, the same value is used for each one. For a bit string, the value specifies the number of bits recorded; for a character string, the value specifies the number of characters recorded.

***DCL:** The string variable is recorded to the end of the string or for a value of 200, whichever is less. If the string variable has a maximum length of zero, only LEN(*DCL) is allowed.

displayed-length: Specify the length of the data recorded. The length (as well as the combination of START and LEN) must be no greater than the length of the shortest string specified in the PGMVAR.

OUTFMT

Specifies the format in which the objects are shown.

***CHAR:** Variables are recorded in character form.

***HEX:** Variables are recorded in both character format and hexadecimal format.

OUTVAR

Specifies whether the values of the program variables are recorded only when their values change, or if they are recorded regardless of any of their values being changed. This parameter is ignored if PGMVAR(*NONE) is specified or assumed.

Note: Within each range, the values of all the traced variables are always recorded the first time a statement in the range is processed. The OUTVAR parameter determines when the variables are recorded for all following statements in the range.

***CHG:** The system records the values of all the program variables when one or more of the values have changed since the last trace point. A variable is considered changed not only when its value is changed, but also when any of the displayed attributes change (such as length, lower/upper bounds, and subscript values). For example, if an array is specified and the upper bound changes for the array, the array is considered to have changed.

Note: The value may not appear to have changed if it contains characters that cannot be shown on the display (a value less than 40 hex). The variable is still recorded even though the user cannot see the change by what is shown. If OUTFMT(*HEX) is specified, the changes can be observed in the traced data.

***ALWAYS:** The system should record the values of the specified variables every time any of the specified trace statements are processed, whether or not any variable had its value changed.

TRCPGM

Specifies the qualified name of the user-supplied, trace-handling program that is called when a statement being traced is reached in the program specified on the PGM parameter. The program with the traced statement passes informational parameters to the trace-handling program when it is called. These parameters identify the program name, the recursion level, the high-level language statement identifier, the machine instruction number at which the traced statement occurred, and a changed variable indicator. The parameters have the following formats:

1. Program name (10 bytes). Specifies the name of the program in which the traced statement was reached.
2. Recursion level (5 bytes). Specifies the recursion level number of the program in which the traced statement was reached. This value is a 1- to 5-digit number padded on the right with blanks.
3. Statement Identifier (10 bytes). Specifies the high-level language program statement identifier that was reached. If the traced statement does not correspond to a statement identifier, the parameter contains a slash (/) followed by a 4-digit hexadecimal machine instruction number.
4. Instruction number (5 bytes). Specifies the machine instruction number that corresponds to the high-level language statement at which the traced statement was reached. No slash appears in front of the machine instruction number. The value consists of 1 to 4 hexadecimal characters representing the MI instruction number, followed by one or more blanks. If the program passes a machine instruction number on the third parameter, the values on the third and fourth parameters will be the same.

All the parameter values are left-justified and padded on the right with blanks. When control returns to the program with the traced statement, processing continues.

When a trace-handling program is specified and OUTVAR(*CHG) is specified, the trace-handling program is called only if a program variable specified on the PGMVAR parameter changes. No trace data is recorded.

***NONE:** No trace-handling program is called when a traced point specified on this command is reached in a batch environment. The interrupted program continues processing.

The name of the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the user-supplied, trace-handling program to be called when a traced statement is reached during debugging in a batch environment. The program specified must not be the same as the program specified on the PGM parameter. If the same program is specified on both the TRCPGM and PGM parameters, results can be unpredictable. After

the program runs, control is returned to the interrupted program and processing continues.

PGM

Specifies the name of the program that contains the specified statement identifiers or the machine instruction numbers that are traced.

***DFTPGM:** The program currently specified as the default program contains the statements to trace.

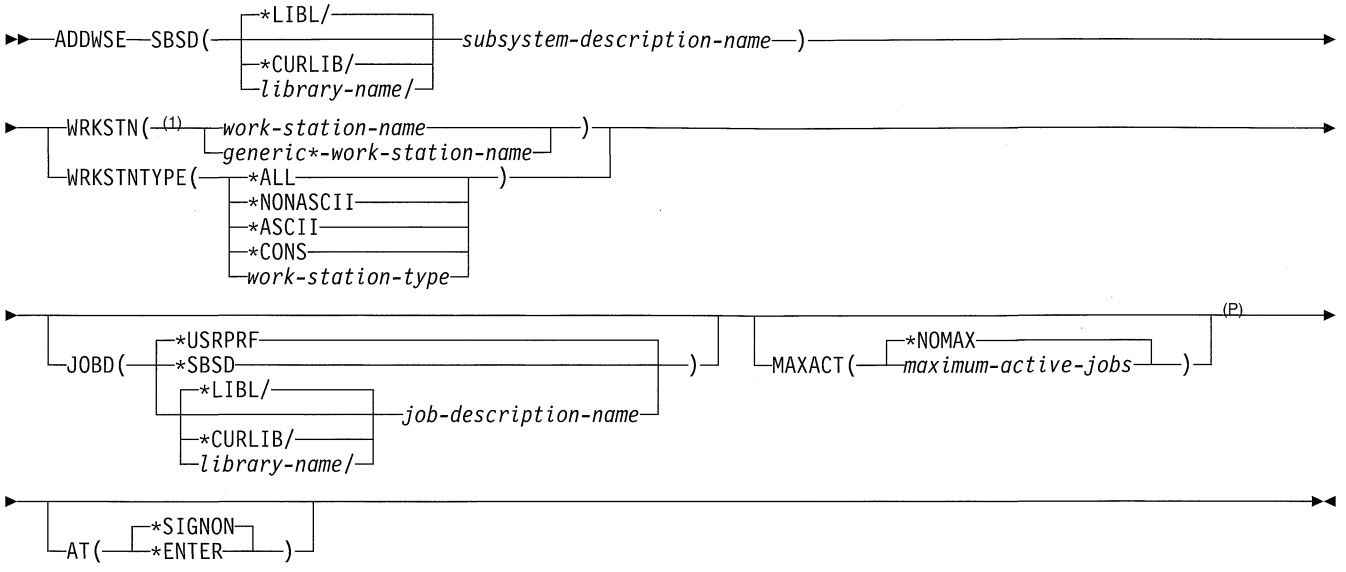
program-name: Specify the name of the program that contains the statements to trace. The specified program must already be in debug mode.

Example

```
ADDTRC   STMT((100 120)  (150 200))
  PGMVAR('&CTR' '&BRCTR' '&SAM')
```

This command traces program statements in the default program between the ranges of statements 100 through 120 and 150 through 200. Also, whenever the values of any of the program variables &CTR, &BRCTR, and &SAM are changed by one of the traced statements within those ranges, the values of all three are recorded before the traced statement is processed. When all of the traced statements have been processed, or when a breakpoint is reached, the Display Trace Data (DSPTRCDTA) command can be used to show the trace data collected.

ADDWSE (Add Work Station Entry) Command



Notes:
1 The WRKSTN parameter and the WRKSTNTYPE parameter are mutually exclusive.
P All parameters preceding this point can be specified in positional form.

Purpose

The Add Work Station Entry (ADDWSE) command adds a work station job entry to the specified subsystem description. The associated subsystem must be inactive at the time. Each entry describes one or more work stations that are controlled by the subsystem. The work stations identified in the work station entries are allowed to sign on the subsystem, or enter the subsystem and run jobs.

Restriction: To use this command, the user must have object operational and object management authorities for the subsystem description.

Required Parameters

SBSD

Specifies the qualified name of the subsystem description to which the work station job entry is added.

Note: The following IBM-supplied objects are not valid on this parameter:

- QLPINSTALL
- QSYSSBSD

The name of the subsystem description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

subsystem-description-name: Specify the name of the subsystem description in which the work station job entry is added.

WRKSTN

Specifies the name of the work station used by the subsystem. The device description name specified in the Create Device Description (Display) (CRTDEVDSP) command associated with the work station is the name that is used.

The WRKSTN parameter and the WRKSTNTYPE parameter are mutually exclusive.

work-station-name: Specify the name of the work station for which a work station entry is added.

generic-work-station-name:* Specify the generic name of the work station. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple

objects can be added only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

WRKSTNTYPE

Specifies the type of work station associated with the entry being added. This entry applies to all work stations of this type that do not have specific entries for an individual work station.

The WRKSTN parameter and the WRKSTNTYPE parameter are mutually exclusive.

The following type codes are valid:

Type Code	Device
3179	3179 Display Station
3180	3180 Display Station
3196	3196 Display Station
3197	3197 Display Station
3277	3277 Display Station
3278	3278 Display Station
3279	3279 Display Station
3476	3476 Display Station
3477	3477 Display Station
3486	3486 Display Station
3487	3487 Display Station
5251	5251 Display Station
5291	5291 Display Station
5292	5292 Color Display Station
5555	5555 Display Station (on systems supporting DBCS (double-byte character set))

***ALL:** The work station entry for all valid work station types is added.

***NONASCII:** The work station entry for all valid work stations that use 5250 data streams is added.

***ASCII:** The work station entries for all work stations that use ASCII data streams are added.

***CONS:** System console display. This entry overrides a device type entry that specifies the same device type as the device being used as the console.

work-station-type: Specify the name of the work station device type for which work station entry is added.

Optional Parameters

JOBD

Specifies the name of the job description used. If the job description does not exist when the entry is changed, a library qualifier must be specified because the qualified job description name is retained in the subsystem description. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***USRPRF:** The job description named in the user profile of the user that signs on at this work station (or at this type of work station) is used for jobs that are started through this entry.

***SBSD:** The job description having the same name as the subsystem description specified by the SBSD parameter is used for jobs created through this entry.

The name of the job description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

job-description-name: Specify the name of the job description used for jobs that are created through this entry.

MAXACT

Specifies, for work stations that use this work station job entry, the maximum number of work station jobs that can be active at the same time. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***NOMAX:** There is no maximum number of jobs that can be active at the same time.

maximum-active-jobs: Specify the maximum number of jobs that can be active at the same time through this work entry.

AT

Specifies when the work stations associated with this job entry are allocated. For more information on how work stations are allocated to subsystems, see the Start Subsystem (STRSBS) command description.

Note: The following should be considered if two or more work station entries specify AT(*SIGNON), if they apply to the same work station and if they are in more than one subsystem description. If the work station is varied on while more than one of the subsystems are active, it cannot be predicted to which subsystem the work station is assigned.

***SIGNON:** The work stations are allocated when the subsystem is started if the work station is not already in use (signed on) in another subsystem. A sign-on prompt is shown at each work station associated with this work entry. If a work station becomes allocated to a different subsystem, interactive jobs associated with the work station are allowed to enter this subsystem through the Transfer Job (TFRJOB) command.

***ENTER:** The work stations associated with this work entry are not allocated when the subsystem is started. However, the interactive jobs associated with the work stations are allowed to enter this subsystem through the Transfer Job (TFRJOB) command.

Examples

Example 1: Adding a Work Station Job Entry

```
ADDWSE  SBSD(LIB7/ORDER)  WRKSTNTYPE(5251)
  JOBD(QCTL)  AT(*SIGNON)
```

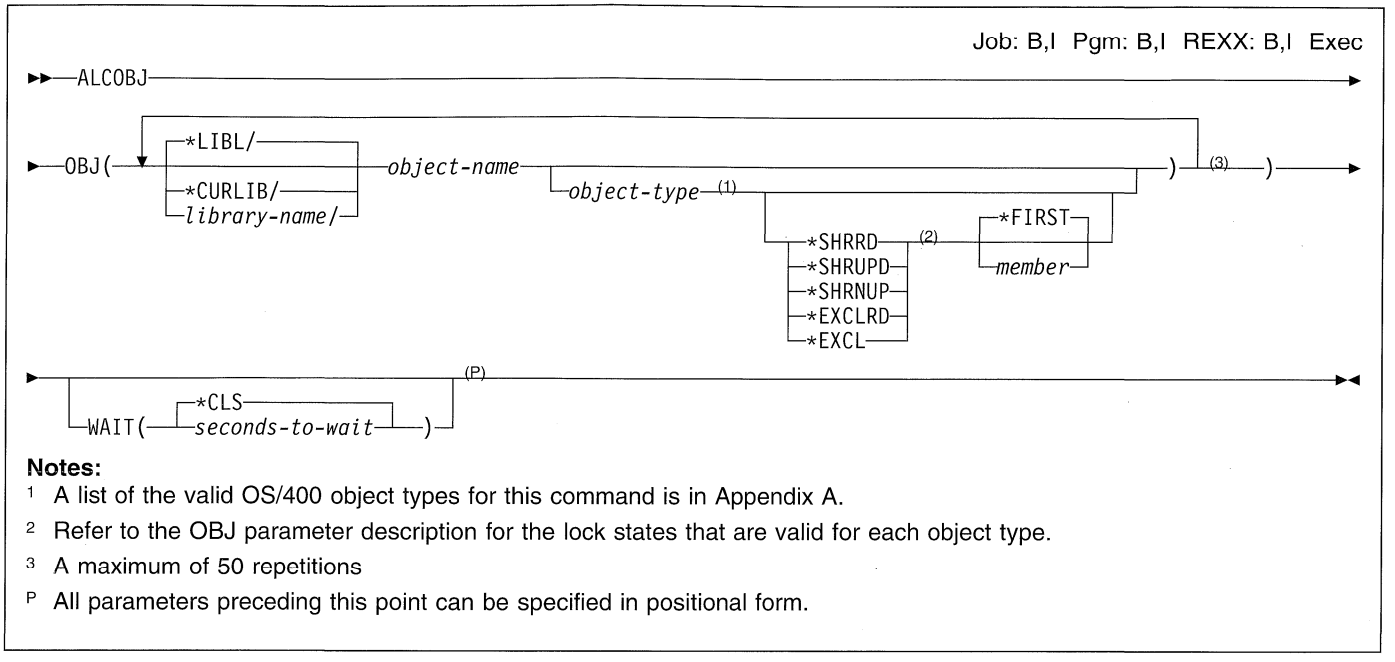
This command adds a work station job entry to a subsystem description named ORDER in library LIB7. All type 5251 work stations are allocated to this subsystem when the subsystem is started, unless they are already active in a previously started subsystem. The work stations are signed on when requested by the user. After sign-on is complete, the IBM-supplied job description QCTL is used to start the routing step.

Example 2: Adding a Work Station Job Entry

```
ADDWSE  SBSD(LIB7/ORDER)  WRKSTN(A12)
  JOBD(LIB7/ORDER)  AT(*SIGNON)
```

This command adds a work station job entry to a subsystem description named ORDER in library LIB7. Work station A12 is signed on when requested by a user.

ALCOBJ (Allocate Object) Command



Notes:

1 A list of the valid OS/400 object types for this command is in Appendix A.

2 Refer to the OBJ parameter description for the lock states that are valid for each object type.

3 A maximum of 50 repetitions

P All parameters preceding this point can be specified in positional form.

Purpose

The Allocate Object (ALCOBJ) command is used in a routing step to reserve an object or list of objects for use later in the routing step. If an object that is needed in the routing step is not specified in an ALCOBJ command, an allocation is made automatically when the object is used. The objects are deallocated either automatically at the end of the routing step or when the Deallocate Object (DLCOBJ) command is used.

The DLCOBJ command should not be issued for an object that was not explicitly allocated by the ALCOBJ command. If the DLCOBJ command is used this way, internal locks on the object are released, making the object capable of being deleted.

Notes:

1. When allocating database files, use the DLCOBJ command before deleting the file if the file being allocated is a logical file.
2. If a file is being allocated that is affected by a file override, the ALCOBJ command ignores the override and attempts to allocate the file named in the OBJ parameter.
3. When allocating distributed data management (DDM) files, additional time is required for the command to complete because of the time required for communication and for allocating files on remote systems.
4. Work station message queues cannot be allocated. A work station message queue is associated with a work station device description of the same name. Therefore, to do an operation on a work station message queue that must be allocated, the user must allocate the associated device description. When the device description is allocated, the work station message queue is implicitly allocated.
5. When ALCOBJ is executed to get an EXCL lock on a program (*PGM), only the program object description is locked. The program code is not locked exclusively. Therefore, the program may still be run by another user. Changes are not allowed for the program object description while the actual program can still be used.

Restrictions:

1. This command cannot be used to allocate a peer device description or device files.
2. The object must exist on the system.
3. The user issuing the command must have object operational authority for the object.
4. The object must not be allocated to another job in a lock state that inhibits or restricts the requested lock state for the entire time specified in the WAIT parameter. If the allocation cannot be completed, none of the locks are granted, and a message is sent to the job that issued the command. If the command is issued from a program, the Monitor Message (MONMSG) command can be used to determine whether the allocation was successful.

Required Parameter

OBJ

Specifies the qualified names of one or more system objects that are allocated to the job, the type of each object specified, the lock state of each object, and, if the object is a database file or DDM file, a member name

can be optionally specified. Refer to the description of the OBJTYPE parameter in Appendix A.

The name of the object can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

Element 1: Name of the Object to be Allocated

object-name: Specify the name of the object that is allocated.

Element 2: Type of Object to be Allocated

object-type: Specify the type of the object to be allocated to the job. Refer to the description of the OBJTYPE parameter in Appendix A.

To determine whether a device description can be allocated, use information from the Work with Configuration Status (WRKCFGSTS) command and from the following table. If, for the appropriate device type, no job name is associated with the device, or if the job name associated with the device is of the same job that issues the ALCOBJ command, and the status field of the display indicates the following value, user can attempt to allocate the device description object.

Table 7. Status of Device Types

Device Type	Status
3179 3180 3196 3197 3277 3278 3279 5555[3] 5251 5291 5292	Vary on pending[1], varied on[2], or sign-on display

Table 7. Status of Device Types

Device Type	Status
3812 4214[1] 4234 4245 5219 5224 5225 5256 5262 5553[3] 5583 *IPDS* 3287 3624 3694 4704 BSC[4] HOST[4] ASYNC[4] SNUF[4] APPC[4] INTRA RETAIL FINANCE NETWORK	Vary on pending[1] or varied on
BSCT 9331 9347 BSC[5]	Varied on

1 Switched device.
2 Device is powered on.
3 Device type is valid only on systems supporting the double-byte character set (DBCS).
4 Not tributary.
5 Tributary.

The device description is not allocated if one of the following conditions exists:

- Another job is allocating the device description
- Another job or object is opening a file to the device
- Another job is varying off the device

If the device description object cannot be allocated, reissue the WRKCFGSTS command to determine the status of the device.

Element 3: Lock States for the Object to be Allocated

lock-state: Specify the lock state for the object. Valid lock states include the following:

Value	Lock State Meaning
*SHRRD	Shared for read
*SHRUPD	Shared for update
*SHRNUP	Shared, no update
*EXCLRD	Exclusive, allow read
*EXCL	Exclusive, no read

Multiple locks can be specified for the same object in the same job with duplicate or different lock states. Each

lock is held separately. For example, if an *EXCL lock is already held for an object, and a second *EXCL lock request occurs, the second lock is acquired. Both locks must be released in the job (deallocated with the DLCOBJ command) before another job can access the same object. If an object is already allocated with one lock state and user want to use a different lock state, first use the ALCOBJ command to request the new lock with the desired lock state and then use the DLCOBJ command to release the old lock (with the old lock state).

Table 8, at the end of this description, shows the system object types that can be specified and the lock states allowed for each object type (A = allowed).

When an exclusive lock is requested on a logical file member, the lock occurs on both the logical file member and the associated physical file members. No other user can use the physical file members (not even through some other logical file member).

The *CL Programmer's Guide* has an explanation of each lock state.

Element 4: Member of the Database File to be Allocated

Note: The following values can only be specified if the object type is a database file.

***FIRST:** The first member of the database file is allocated to the job.

member: Specify the name of the member to be allocated to the job. If the specified file is a logical file, the physical file members associated with the members of the logical file are also allocated to the job.

Optional Parameter

WAIT

Specifies the number of seconds that the program waits for the object to be allocated. If the object cannot be allocated in the specified wait time, a message, which can be detected by a Monitor Message (MONMSG) command, is sent to the program. If one or more device descriptions are in the list of objects to be allocated, the system may wait more than the specified amount of time to attempt the allocation.

When allocating DDM files, additional time is required for communications and for allocating files on remote systems. A separate WAIT time is used for each remote system.

***CLS:** The job default wait time is used as the wait time for the file resources being allocated.

Table 8. Lock States

Object Type	*EXCL	*EXCLRD	*SHRUPD	*SHRNUP	*SHRRD
*AUTL	A	A	A	A	A
*DEVD[1]		A			
*DTAARA	A	A	A	A	A
*DTADCT	A	A	A	A	A
*DTAQ	A	A	A	A	A
*FCT	A	A	A	A	A
*FILE[2]	A	A	A	A	A
*LIB		A	A	A	A
*MENU	A	A	A	A	A
*MSGQ	A				A
*PNLGRP	A	A	A	A	A
*PGM	A	A			A
*SBSD	A				
*SCHIDX	A	A	A	A	A
*SSND	A	A	A	A	A
*S36	A	A	A	A	A

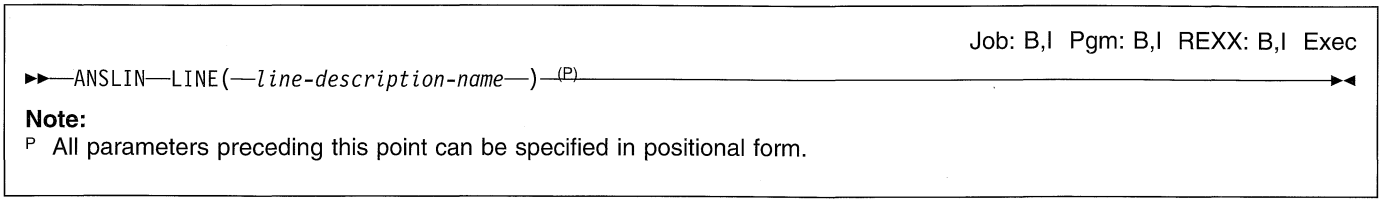
1 APPC and INTRA device descriptions cannot be allocated on the ALCOBJ command.
2 Only database file members can be allocated.

Example

```
ALCOBJ   OBJ((LIBB/FILEA  *FILE *EXCL MEMBERA))
```

This command exclusively allocates MEMBERA of FILEA in LIBB to the routing step in which the ALCOBJ command is used. If MEMBERA is unavailable, the number of seconds to wait for it to become available is the default wait time defined for the class used by the routing step.

ANSLIN (Answer Line) Command



Purpose

The Answer Line (ANSLIN) command identifies a communications line that is manually answered by the system operator. This command indicates that the operator manually answered an incoming call and validated the requirements of the caller. When this command is entered, the manual answer sequence is run for the line and, when completed, instructs the operator to select data mode on the modem.

Required Parameter

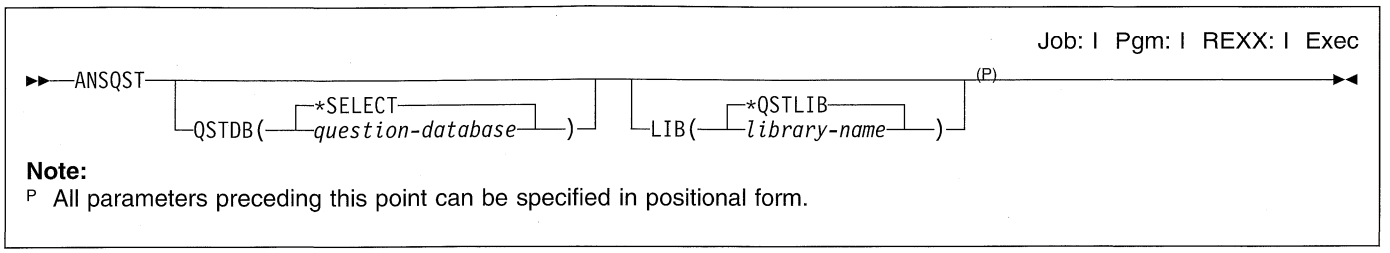
LINE
Specifies the name of the communications line that is being answered.

Example

```
ANSLIN  LINE(LINE01)
```

This command answers an incoming call on a line named LINE01.

ANSQST (Answer Questions) Command



Note:
P All parameters preceding this point can be specified in positional form.

Purpose

The question-and-answer (Q & A*) database coordinator uses the Answer Questions (ANSQST) command to display and answer questions asked by users of a Q & A database. More information is in the *Q & A Database Coordinator's Guide*.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. A user must have authority to the command and be a Q & A coordinator for any Q & A database referred to by the command.
3. This command is interactive only.

Optional Parameters

QSTDB

Specifies the Q & A database with which to show and answer questions.

***SELECT:** The user is asked to specify a Q & A database. If only one Q & A database exists on the system, it is the default.

question-database: Specify the name of the Q & A database with which to display and answer questions.

LIB

Specifies the name of the library that contains the Q & A database.

***QSTLIB:** The library containing the specified Q & A database is searched. If *SELECT is specified on the QSTDB parameter, any Q & A database in any library for which the user is authorized can be selected.

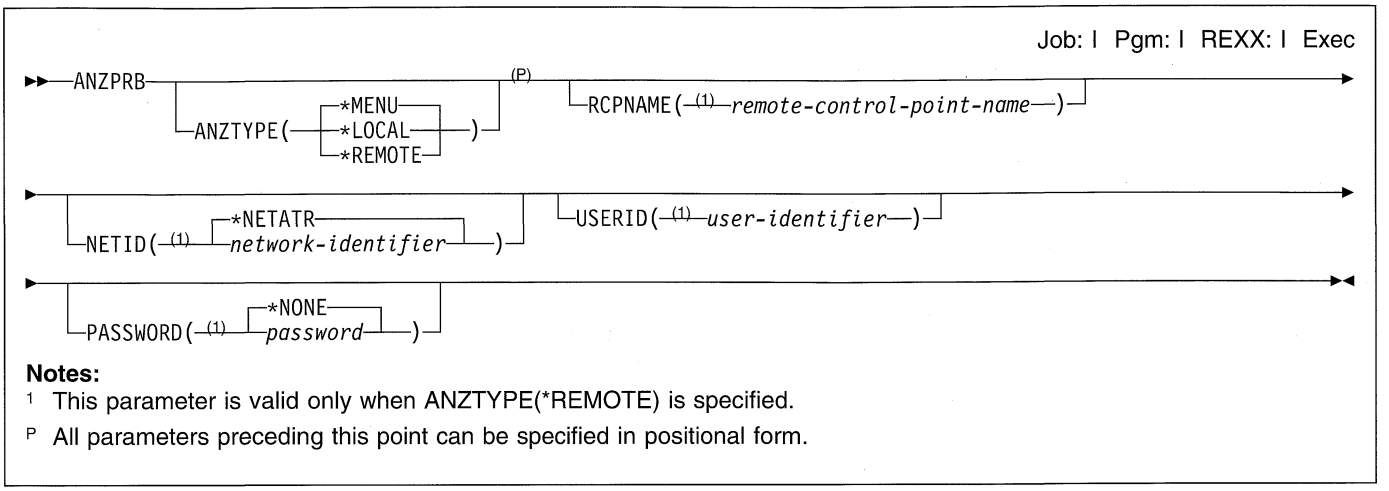
library-name: Specify the name of the library to be searched. If *SELECT is specified on the QSTDB parameter, any database in the library for which the user is authorized can be selected.

Example

```
ANSQST
```

This command shows the Select Question Status display. If more than one Q & A database is available for selection, the Select Q and A Database display is shown first.

ANZPRB (Analyze Problem) Command



Notes:
1 This parameter is valid only when ANZTYPE(*REMOTE) is specified.
P All parameters preceding this point can be specified in positional form.

Purpose

The Analyze Problem (ANZPRB) command allows you to analyze, create problem records for, or report problems that are not detected by the system. For example, you can analyze or report:

- Job or programming problems.
- Equipment or communications problems.
- Problems that made it necessary to do an initial program load (IPL) of the system again.
- Problems on a device or system not attached to your system.

Restriction: This command is shipped with public *EXCLUDE authority and the QPGMR, QSYSOPR, QSRV, and QSRVBAS user profiles have private authorities to use the command.

Optional Parameters

ANZTYPE
Specifies the type of analysis to do. The default value, *MENU, allows you to do a local or remote analysis on an AS/400 system or to do a remote analysis on another system that is not an AS/400 system. If you want to do an analysis on a local or remote AS/400 system, then you can bypass the initial menu by specifying *LOCAL or *REMOTE.

Note: The user cannot do remote analysis unless the AS/400 Systems Management Utilities (5730SM1) licensed program is installed.

***MENU:** The *Select Type of System* menu is shown. You can analyze problems on:

- The AS/400 system.
- Another AS/400 system which is enrolled as a service requester.
- Another AS/400 system which is not enrolled as a service requester.
- Another system which is not an AS/400 system.

***LOCAL:** Local analysis is selected. Problem analysis is done on the AS/400 system.

***REMOTE:** Remote analysis is selected. Problem analysis is done for another AS/400 system that is enrolled as a service requester.

RCPNAME
Specifies the remote control point name for the service requester system where the remote analysis is done.

NETID
Specifies the network identifier (ID) for the service requester system where the remote analysis is done.

***NETATR:** The network ID of the service provider is used.

network-identifier: Specify the network ID.

USERID
Specifies the user identifier (ID) used to access the remote system.

PASSWORD
Specifies the password used to access the remote system.

Note: This field may be left blank if the service requester system is an unsecured system.

***NONE:** No password is needed to access the remote system because the remote system has a security level of 10.

password: Specify the password used to access the remote system.

Examples

Example 1: Displaying the Menu

```
ANZPRB
```

This command shows the Analyze Problem menu.

Example 2: Starting Remote Analysis

```
ANZPRB  ANZTYPE(*REMOTE)
```

This command shows the display which prompts for the remaining values of the command. After you specify the appropriate values, remote analysis begins.

Example 3: Accessing Remote System with User ID and Password

```
ANZPRB  ANZTYPE(*REMOTE)  RCPNAME(RCH38377)
  USERID(JON)  PASSWORD
```

This command shows the display which prompts for the remaining values of the command. After you specify the appropriate values beyond the ones specified on the command example, remote analysis begins.

Example 4: Remote Analysis has Security Level of 10

```
ANZPRB  ANZTYPE(*REMOTE)  RCPNAME(RCH38377)
  USERID(JON)
```

This command is slightly different than the preceding example. The same display prompt appears; however, if PASSWORD is not specified, the system assumes that the remote system has a security level of 10, that is, it does not use passwords. After you specify the appropriate values beyond the ones specified on the command example, remote analysis begins.

Example 5: Displaying Menu

```
ANZPRB  ANZTYPE(*MENU)
```

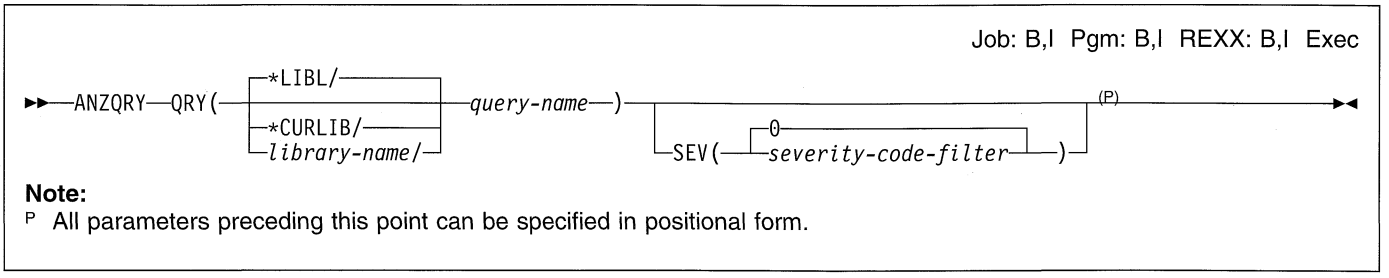
This command shows a menu prompting you for the type of analysis to be done. The remaining parameters do not appear on the display.

Example 6: Starting Local Analysis

```
ANZPRB  ANZTYPE(*LOCAL)
```

This command begins analysis on the local device. The remaining parameters do not appear on the display.

ANZQRY (Analyze Query) Command



Note:

P All parameters preceding this point can be specified in positional form.

Purpose

The Analyze Query (ANZQRY) command allows the user to analyze a query definition (QRYDFN) object for query management conversion problems. Output from this command includes diagnostic messages about potential differences between Query/400 and query management use of query and form information derived from the analyzed QRYDFN object. A completion message shows the highest severity of potential problems that are found.

Required Parameter

QRY

Specifies the name of the query definition (QRYDFN) to be analyzed.

The name of the QRYDFN can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

query-name: Specify the name of the QRYDFN to be analyzed.

Optional Parameter

SEV

Specifies the severity code of the message. The severity code indicates the severity level of the condition that causes the message to be sent.

0: All diagnostic messages about differences are logged.

severity-code-filter: Specify a severity code filter. Valid values range from 0 through 99.

Examples

Example 1: Displaying All Messages

```
ANZQRY  QRY2
```

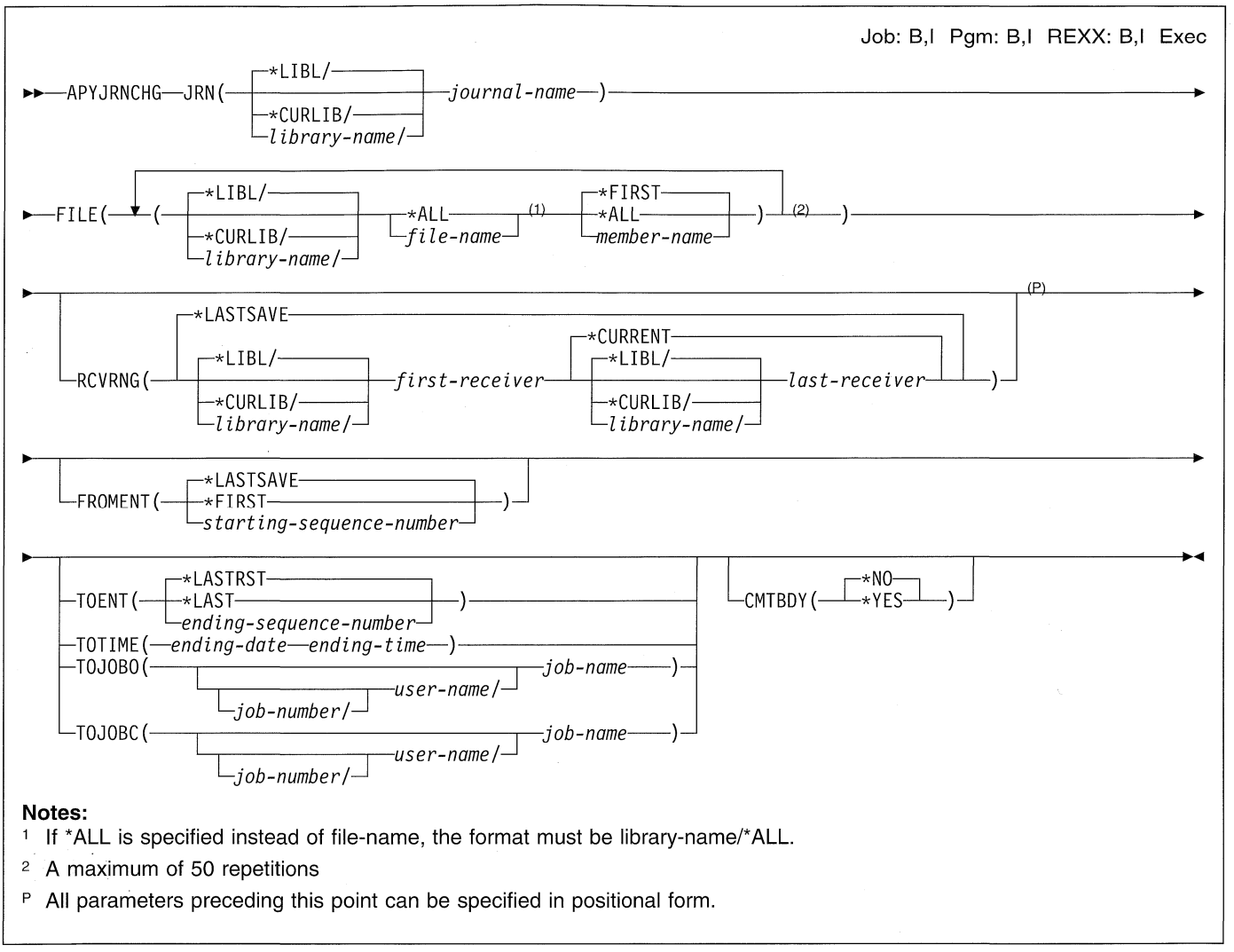
This command analyzes the first QRYDFN named QRY2 in the user's library list. Messages about conversion problems, for example, text that is too long, are sent to the job log. The messages are displayed when the analysis has completed.

Example 2: Displaying Specific Messages

```
ANZQRY  QRY2 99
```

This command analyzes the first query named QRY2 in the user's library list. Only the completion message and messages diagnosing conditions which need to be investigated before a run is attempted are shown and logged.

APYJRNCHG (Apply Journaled Changes) Command



Notes:

1 If *ALL is specified instead of file-name, the format must be library-name/*ALL.

2 A maximum of 50 repetitions

P All parameters preceding this point can be specified in positional form.

Purpose

The Apply Journaled Changes (APYJRNCHG) command applies the changes that have been journaled (for a particular member of a database file) to a saved version of the file to recover the file after an operational error or some form of damage. The journaled changes are applied from the specified starting point, either the point at which a file was last saved or a particular entry on the journal, until the specified ending point has been reached. The ending point can be the point at which the file has had all changes applied, the file was last restored, a specified entry has been reached, an assigned time has been reached, or the file was opened or closed by a job (the CMTBDY parameter is used for handling changes that are still pending in the file).

Note: The Display Journal (DSPJRN) command can be used to help determine the desired starting and/or ending points.

A list of physical files and members can be specified. The journaled changes for physical file members are applied in the order that the journal entries are found on the journal, which is the same order in which the changes are made to the physical file members.

If an error is found at any point when applying journaled entries, the command ends and the file members may be only partially changed from the journal entries. The following is a partial list of errors:

- The member was reorganized
- The member was restored
- Journaling was stopped for the member
- The member was deleted or saved with storage freed
- Journal IPL synchronization fails
- Partial damage to a journal receiver
- Logical errors in a file member such as a duplicate key
- The member had its changes applied or removed (through the APYJRNCHG command or the Removed Journaled Changes (RMVJRNCHG) command).

The command also ends on illogical conditions, such as trying to add a record to an existing relative record number,

add a record beyond the next record position after the end of the file, delete a deleted record, and update a nonexistent record.

The user of the command may issue the command again, specifying a new starting sequence number, if it is logically possible to restart the apply operation.

It is possible to apply changes even if the sequence numbers have been reset. The system sends an informational message and continues to apply the changes. If journal receivers are attached and detached in pairs (dual receivers), the system tries to use the first of the two receivers (the first of the two shown in the WRKJRNA receiver directory). When the first of the pair cannot be used (for example, is damaged or not found), the system tries to use the second receiver of the pair. If neither receiver is usable, the attempts to apply changes end and no changes are applied.

Restrictions:

1. This command is shipped with public *EXCLUDE authority and the QPGMR and QSRV user profiles have private authorities to use the command.
2. The files specified on this command must currently have their changes journaled and they must have been journaled to the specified journal throughout the period indicated on the command.
3. If a restore operation occurs before the apply operation, the file being restored must have been journaled at the time of the save operation. The files indicated on the command are allocated exclusively while the changes are being applied.
4. If a file cannot be allocated, the command ends and no journaled changes are applied.
5. If there is no journal entry that corresponds to the period indicated on the command, the command ends and no journaled changes are applied.
6. If the journal sequence numbers are reset in the range of the receivers specified and the parameters are specified, the first occurrence of the FROMENT or TOENT parameter is used.

Note: If applying journaled changes ends for one of the members specified, it ends for all of the members specified.

Required Parameters

JRN

Specifies the qualified name of the journal associated with the journal entries that are applied.

The name of the journal can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

journal-name: Specify the name of the journal associated with the journal entries being applied.

FILE

Specifies a maximum of 50 qualified names of physical database files to which journal entries are being applied.

The name of the file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

Element 1: File Name

***ALL:** All physical files in the specified library whose changes are journaled to the specified journal have their journal entries applied. The library name must be specified. If *ALL is specified and the user does not have the required authority for all the files in the library, a message is sent and the applying of journal entries ends.

file-name: Specify the name of the physical database file that is to have its journal entries applied.

Element 2: Member Name

The FILE parameter also specifies the name of the member in the file that has its journal entries applied.

***FIRST:** The first member in the file has its journal entries applied.

***ALL:** All members in the file have their journal entries applied.

member-name: Specify the name of the member in the file that has its journal entries applied.

If *ALL is specified for the first part of this parameter, the value specified for the member name is used for all applicable files in the library. For example, if *FIRST is specified, the first member of all applicable files in the library has the changes applied.

Note: The APYJRNCHG command can apply changes to a maximum of 1024 members. If more than 1024 members are specified, an error message is sent and no changes are applied. The values entered on the FILE parameter must be changed so that the limit is not exceeded.

Optional Parameters

RCVRNG

Specifies the first and last journal receivers used in applying the journal entries. The system begins by applying the first journal receiver (specified by the first value) and proceeds through the receivers until it applies the last receiver (specified by the last value). If dual receivers are used at any time, the first of the receivers is always used when chaining through the set of receivers. If any problem is found in the receiver chain (such as a damaged receiver or a receiver not found) before the journal entries are applied, the system tries to use the second of the dual receivers. If the second of the receivers is damaged or not found, or if the problem is found when applying journal entries, the operation ends.

***LASTSAVE:** The range of journal receivers used is determined by the system, as a result of save information for the files that have their journaled changes applied. This parameter value is only valid if FROMENT(*LASTSAVE) is also specified.

***CURRENT:** Only the currently attached receiver is used in applying the journal entries.

Element 1: First Journal Entry

The name of the journal receiver can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

first-receiver: Specify the name of the journal receiver used as the first (oldest) receiver.

Element 2: Second Journal Entry

***CURRENT:** The applying of journal entries continues for all journal receivers in the chain, beginning with the receiver specified by the first parameter value through the currently attached journal receiver.

Note: The maximum number of receivers that can be used in a range of receivers is 256. If more than 256 receivers are specified, an error message is sent and no changes are applied.

The name of the journal receiver can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

last-receiver: Specify the name of the journal receiver used as the last (newest) receiver with journal entries to be applied. If the end of the receiver chain is reached before finding this receiver, no entries are applied, and an escape message is sent.

FROMENT

Specifies the entry that is used as the starting point for applying changes that have been journaled.

***LASTSAVE:** The journal entries are applied beginning with the first journal entry after the file member was last saved. The system determines the actual starting position for each of the file members specified on the command. The parameter value implies that the file was just restored on the system.

The system verifies information for each member specified, such as if the date and time of the restore is after the date and time of the last save. The system also verifies that the date and time of the saved version of the file member that is restored on the system is the same as the date and time that the file member was last saved, as indicated on the journal.

If the dates and times do not match, no entries are applied and an inquiry message is sent to the system operator requesting a cancel or ignore response. If an ignore response is given to the message, the operation is attempted. A cancel response causes the operation to end, and no journal entries are applied.

If the file was last saved with the save-while-active function, the saved copy of each file member includes all record-level changes in the journal entries up to the corresponding 'F SS' start of save journal entry. In this case, the system applies changes beginning with the first journal entry following the 'F SS' entry.

If the file was last saved when it was not in use (normal save), the saved copy of each member includes all record-level changes in the journal entries up to the corresponding 'F MS' member saved journal entry. In this case, the system applies changes beginning with the first journal entry following the 'F MS' entry.

***FIRST:** The journal entries are applied beginning with the first journal entry in the first receiver supplied in this command.

starting-sequence-number: Specifies the sequence number of the first journal entry that is applied from the journal entries supplied.

TOENT

Specifies the entry used as the ending point for applying changes that are journaled.

***LASTRST:** The journal entries are applied ending with the entry before the file member was last restored. The system determines the actual ending position for each of the file members specified on the command. The system verifies that the date and time of the restored

version of the file member on the system is the same as the date and time that the file member was last restored, as indicated on the journal. If the dates and times do not match, no entries are applied and an inquiry message is sent to the user or system operator, requesting a cancel or ignore response. If an ignore response is given to the message, the operation is attempted. A cancel response causes the operation to end, and no journal entries are applied.

This parameter value is valid only if FROMENT(*LASTSAVE) is also specified. If no other TO parameter (TOTIME, TOJOBO, TOJOBC) is specified, TOENT(*LASTRST) is assumed.

***LAST:** Journal entries are applied through the last entry.

ending-sequence-number: Specify the sequence number of the last entry that is applied to the file member.

TOTIME

Specifies the time and date of the last journal entry that was applied to the file member. The first entry with that or the next earlier time is the ending point for applying journal entries. The format of the date must be defined by the job attributes DATFMT and, if separators are used, DATSEP. The time can be entered as 4 or 6 digits (hhmm or hhmmss), where hh = hours, mm = minutes and ss = seconds. If colons are used to separate the time values, the string must be enclosed in apostrophes ('hh:mm:ss').

Element 1: Ending Date

ending-date: Specify the ending date.

Element 2: Ending Time

ending-time: Specify the ending time.

TOJOBO

Specifies that the journal entries are only applied until the indicated job (qualified job name) first opens any physical file member (or logical member defined over the physical member) in the list of members specified on the FILE parameter that have their journal entries applied. (This is the ending point for all members specified.)

A job identifier is a special value or a qualified name with up to three elements. For example:

```
job-name
user-name/job-name
job-number/user-name/job-name
```

Job name must be specified. *N may be used in place of job number or user name to maintain the position in the parameter value sequence. For example, 123456/*N/"job-name" specifies the job number, 123456, and the job name. It does not specify user name.

job-name: Specify the name of the job.

user-name: Specify the name of the user of the job.

job-number: Specify the number of the job.

TOJOBC

Specifies that the journal entries are only applied until the indicated job (qualified job name) last closes any physical file member (or logical member defined over the physical member) that is in the list of members specified on the FILE parameter that have their journal entries applied, or until the indicated job is ended. (This is the ending point for all members specified.)

Note: See the TOJOBO parameter for a description of how qualified job names are specified.

job-name: Specify the name of the job.

user-name: Specify the name of the user.

job-number: Specify the number of the job.

CMTBDY

Specifies the method by which pending changes are handled when journal changes are applied to database files in the commitment control environment. More information on the use of commitment control is in the *Advanced Backup and Recovery Guide*.

***NO:** The journal entries are applied from the specified FROMENT parameter to the specified TO option regardless of where the commitment boundaries are located. If the FROM or TO option identifies an entry that is in the middle of a commit cycle, the operation is attempted.

***YES:** The journal entries are applied from the specified FROMENT parameter to the specified TO option, honoring commitment boundaries. If the specified FROMENT parameter is in the middle of a commit cycle, an error message is sent and the operation is not attempted. If the specified TO option is in the middle of a commit cycle, the operation stops at the commitment boundary that first comes before the specified TO option. When this occurs, a diagnostic message is sent at the end of the operation.

Examples

Example 1: Applying Changes to First Member

```
APYJRNCHG   JRN(FIN/JRNACT)  FILE(FIN/RCVABLE)
```

This command causes the system to apply to the first member of file RCVABLE in library FIN all changes journaled to JRNACT in library FIN since the file was last saved. The receiver range is determined by the system. The changes are applied beginning with the first journaled change on the receiver chain after the file was last saved and continue through all applicable journal entries to the point at which the file was last restored.

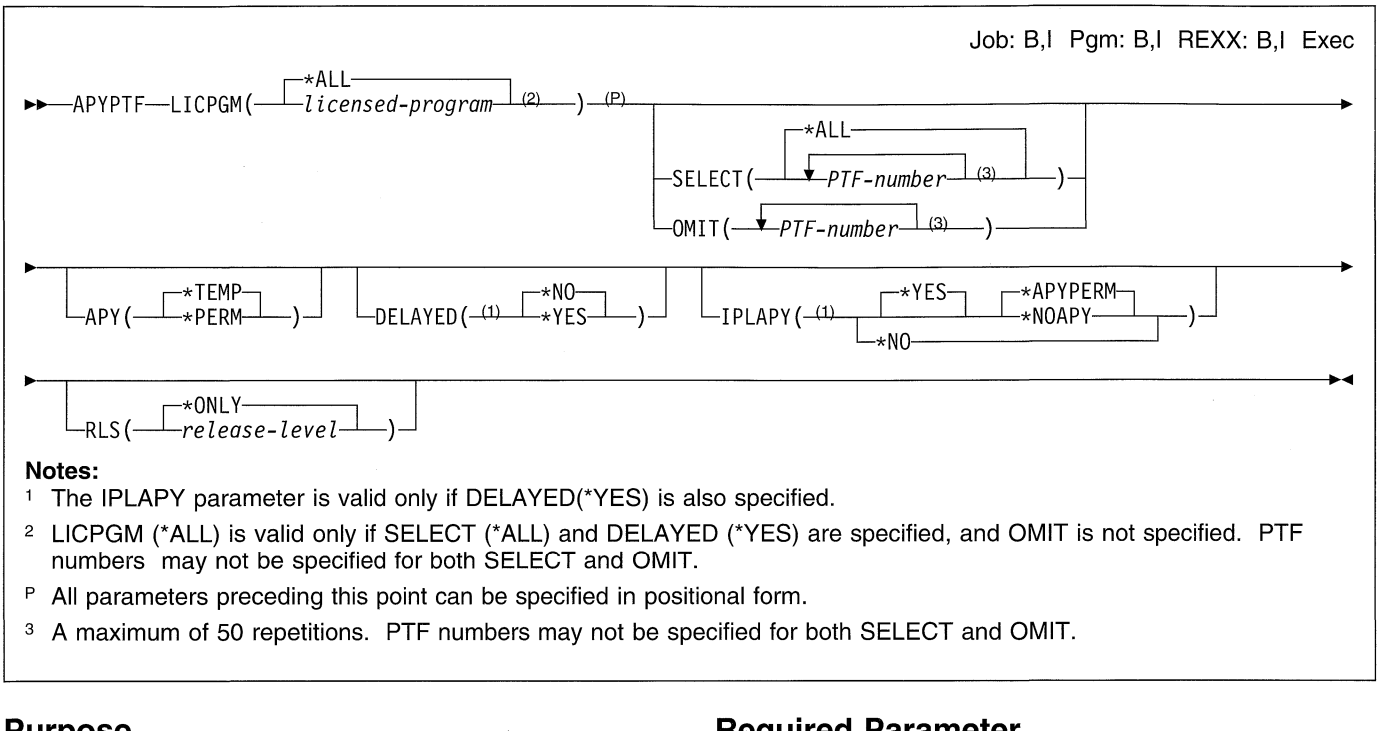
Example 2: Applying Changes to a Specific Member

```
APYJRNCHG   JRN(JRNA)  FILE((LIB2/PAYROLL JAN))
  RCVRNG(RCV22 RCV25)  FROMENT(*FIRST)  TOENT(*LAST)
```

This command causes the system to apply all changes journaled to JRNA to member JAN of the file PAYROLL in library LIB2. The journal receivers containing the journaled changes

are contained in the receiver chain starting with receiver RCV22 and ending with receiver RCV25. Applying the changes starts with the first change journaled on this receiver chain and ends with the last change journaled on this receiver chain. The library search list (*LIBL) is used to find the journal JRNA and the journal receivers RCV22 and RCV25.

APYPTF (Apply Program Temporary Fix) Command



Notes:

1 The IPLAPY parameter is valid only if DELAYED(*YES) is also specified.

2 LICPGM (*ALL) is valid only if SELECT (*ALL) and DELAYED (*YES) are specified, and OMIT is not specified. PTF numbers may not be specified for both SELECT and OMIT.

P All parameters preceding this point can be specified in positional form.

3 A maximum of 50 repetitions. PTF numbers may not be specified for both SELECT and OMIT.

Purpose

The Apply Program Temporary Fix (APYPTF) command applies Program Temporary Fixes (PTFs) to a specified licensed program. Before a PTF can be applied, it must be loaded by the Load Program Temporary Fix (LODPTF) command.

When a PTF is applied, it completely replaces the affected objects in the licensed program. PTFs can be applied temporarily or permanently. If they are applied temporarily, the replaced objects are saved by the system and can later be restored to the program by the Remove Program Temporary Fix (RMVPTF) command. If PTFs are applied permanently, the replaced objects are deleted from the system.

The APYPTF command is used to apply immediate PTFs at the time the command is run, or to request PTFs to be applied during the next unattended initial program load (IPL).

During an attended IPL, the Work with PTFs display is used to apply PTFs at the time the system is started. Some IPLs may take longer than others when PTFs are being applied. More information on applying PTFs is in the *Operator's Guide*.

Restriction: This command is shipped with public *EXCLUDE authority and the QSRV user profile has private authority to use the command.

Required Parameter

LICPGM

Specifies the 7-character identifier of the licensed program for which the PTFs are to be applied.

Note: LICPGM(*ALL) is valid only if SELECT(*ALL) and DELAYED(*YES) are specified, and OMIT is not specified on this command.

***ALL:** PTFs are applied to all installed licensed programs, PRPQs, and LPOs.

licensed-program: Specify the 7-character licensed program identifier to which PTFs are applied.

Optional Parameters

SELECT

Specifies which of the previously loaded PTFs are being applied to specified licensed programs. The OMIT parameter cannot be specified if single PTF numbers are specified in the SELECT parameter.

***ALL:** All the PTFs that were loaded are being applied to the program. If all PTFs cannot be applied, messages are sent indicating the PTFs that were not applied and the reasons (for example, required PTFs were not yet applied).

PTF-number: Specify the PTF identification number of each programming fix being applied. Up to 50 PTF numbers can be specified.

OMIT

Specifies the PTF numbers that are not applied. Up to 50 PTF numbers can be specified. The OMIT parameter

cannot be specified if single PTF numbers are specified in the SELECT parameter.

APY

Specifies whether the PTFs are applied on a temporary or permanent basis. Permanently applied fixes cannot be removed; temporarily applied fixes can be removed by the Remove Temporary Fix (RMVPTF) command.

***TEMP:** The PTFs are applied as temporary PTFs.

***PERM:** The PTFs are applied permanently.

DELAYED

Specifies whether immediate PTFs are applied at the time the command is run, or whether immediate or delayed PTFs are applied during the next unattended IPL.

***NO:** Immediate PTFs that are identified are applied at the time the command is processed. Delayed PTFs are ignored during the APYPTF request and are not applied. A message is sent for each PTF that is not applied.

***YES:** Both delayed and immediate PTFs that are identified are applied during the next unattended IPL. The IPLAPY parameter determines whether the PTFs are applied during the next unattended IPL, or whether any request to apply PTFs during the next unattended IPL is canceled.

IPLAPY

Specifies the action that is done for delayed or immediate PTFs at the next unattended IPL. This parameter is valid only if DELAYED(*YES) is also specified.

Element 1: PTFs Applied at IPL

***YES:** The identified PTFs are applied at the next unattended IPL. The APY parameter determines whether the apply operation is temporary or permanent.

Element 2: Licensed Internal Code Fixes Applied

***APYPERM:** If the licensed program PTFs applied during an unattended initial program load (IPL) have prerequisites to Licensed Internal Code fixes, then the required Licensed Internal Code fixes are also identified to be permanently applied during the unattended IPL.

***NOAPY:** No Licensed Internal Code prerequisite PTFs are identified to be applied.

Single Value

***NO:** Previous requests to apply the identified PTFs at the next unattended IPL are canceled.

RLS

Specifies the release level of the software product. If multiple releases are installed, the release is required.

***ONLY:** This value is valid only when *one* release is installed on the system.

release-level: Specify the release level in the format VxRxMx, where Vx is the version number, Rx is the release number, and Mx is the modification number.

Examples

Example 1: Applying Temporary PTFs

```
APYPTF  LICPGM(5738SS1)  DELAYED(*YES)
```

This command applies all the programming fixes that affect Operating System/400 (licensed program 5738-SS1). The fixes are temporarily applied at the next IPL.

Example 2: Applying Permanent PTFs

```
APYPTF  LICPGM(5738SS1)  SELECT(SF00003 SF00008 SF00012)
  APY(*PERM)  DELAYED(*YES)
```

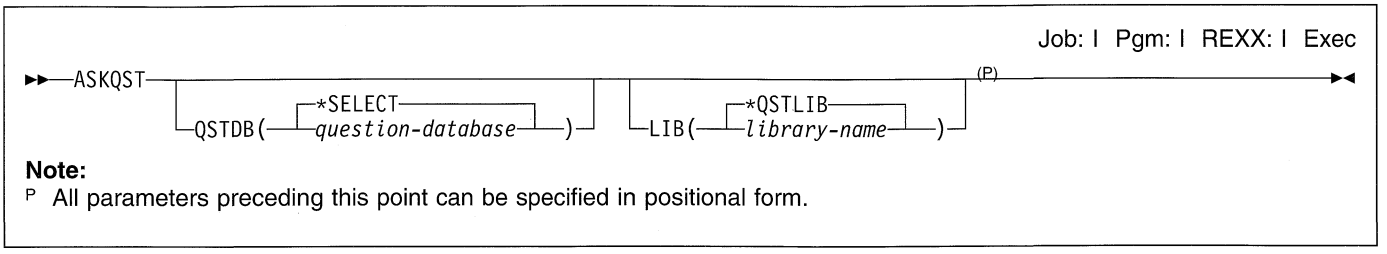
This command permanently applies PTFs SF00003, SF00008, and SF00012 to the Operating System/400 in library QSYS at the next IPL.

Example 3: Applying All Loaded PTFs

```
APYPTF  LICPGM(*ALL)
  DELAYED(*YES)
```

This command permanently applies all PTFs to the Operating System/400 in library QSYS at the next IPL. The fixes are temporarily applied at the next IPL.

ASKQST (Ask Question) Command



Note:
| P All parameters preceding this point can be specified in positional form.

Purpose

The Ask Question (ASKQST) command shows the Search for Answers display; from this display the user can search for an answer to a question. The user must first search the database to determine if an answer exists before a question can be asked. More information is in the *Q & A Database Coordinator's Guide*

Optional Parameters

QSTDB

Specifies the Question-and-Answer (Q & A) database in which to ask a question.

***SELECT:** The user is asked to specify a Q & A database. If only one Q & A database exists on the system, it is the default.

question-database: Specify the name of the Q & A database in which to ask a question.

LIB

Specifies the name of the library that contains the Q & A database.

| The name of the Q & A database can be qualified by one of the following library values:

***QSTLIB:** The library containing the specified Q & A database is used. If *SELECT is specified on the QSTDB parameter, any Q & A database for which the user is authorized can be selected.

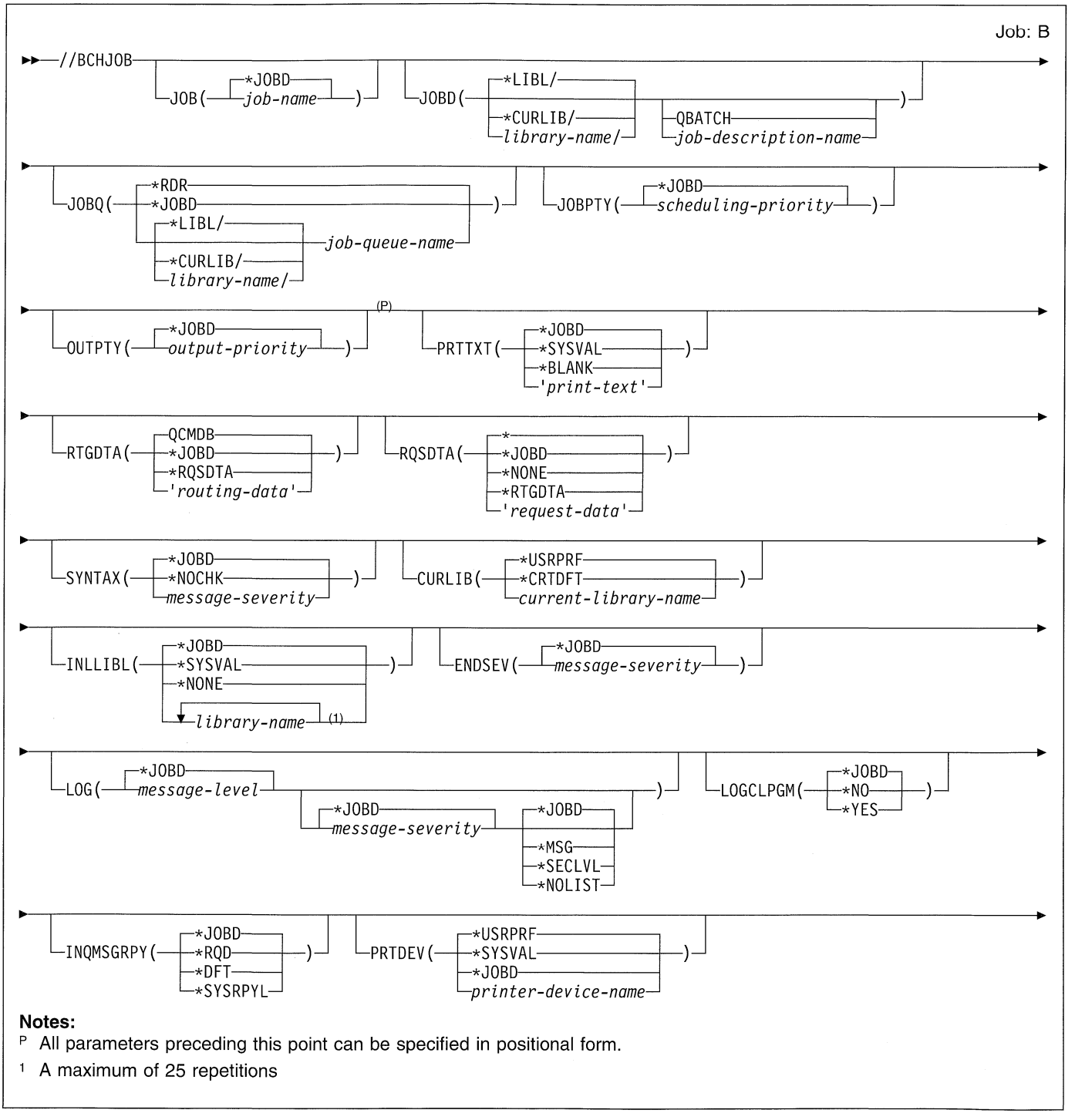
library-name: Specify the name of the library to be searched. If *SELECT is specified on the QSTDB parameter, any database for which the user is authorized can be selected.

Example

```
ASKQST
```

This command shows the Search for Answers display.

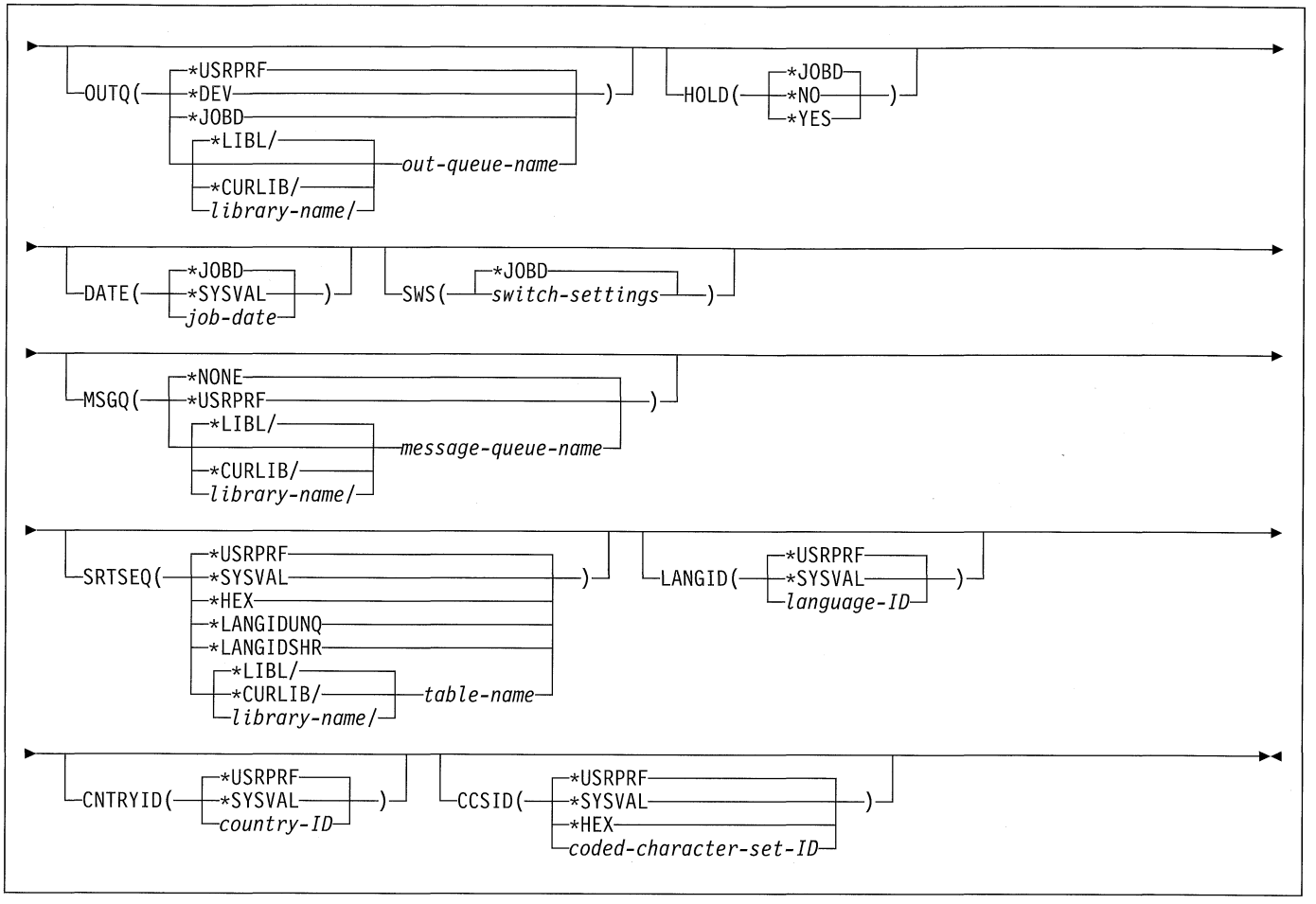
BCHJOB (Batch Job) Command



Notes:

P All parameters preceding this point can be specified in positional form.

1 A maximum of 25 repetitions



Purpose

The Batch Job (//BCHJOB) command indicates the beginning of a batch job in a batch input stream. It can also specify different values for the attributes for the job instead of the ones specified in the job description used with this job. The values contained in the job description are used for parameters not coded in the BCHJOB command.

Restrictions:

1. The BCHJOB command cannot be used from a work station.
2. Two slashes must precede this command name when entering it in the data record:

   ```
   //BCHJOB
   ```

   The user can separate the slashes from this command name with blank spaces, for example, `//  BCHJOB`.

Optional Parameters

JOB

Specifies the name associated with the job when it is processed by the system.

***JOBD:** The simple name of the job description used with this job is the name of the job itself.

job-name: Specify the job's simple name used during processing.

JOBD

Specifies the qualified name of the job description used with this job.

The name of the job description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

QBATCH: The IBM-supplied job description, QBATCH, in the QGPL library is used for the job. (The QGPL library must be in the library list used by the spooling reader that reads the job's input.

job-description-name: Specify the qualified name of the job description.

JOBQ

Specifies the qualified name of the job queue on which this job is placed.

***RDR:** The job queue specified in the start reader or submit jobs command that reads this job is the job queue used.

***JOBD:** The job queue named in this job description is used.

The name of the queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

job-queue-name: Specify the qualified name of the job queue on which the submitted job is placed.

JOBPTY

Specifies the scheduling priority of the job. Valid values range from 1 through 9, where 1 is the highest priority and 9 is the lowest priority. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***JOBD:** The scheduling priority specified in the job description is used for this job.

scheduling-priority: Specify a value, ranging from 1 through 9, that is the scheduling priority for this job.

OUTPTY

Specifies the output priority for spooled files that are produced by this job. The highest priority is 1 and the lowest priority is 9. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***JOBD:** The output priority specified in the job description is used for this job.

output-priority: Specify a value, ranging from 1 through 9, for the priority of this job's output files.

PRTTXT

Specifies up to 30 characters of text to be printed at the bottom of each page of output. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***JOBD:** The value specified in the job description is used.

***SYSVAL:** The print text is obtained from the system value QPRTTXT.

***BLANK:** Text is not specified.

'print-text': Specify the character string that is printed at the bottom of each page. Up to 30 characters can be entered, enclosed in apostrophes if necessary.

RTGDTA

Specifies the routing data used to start the first routing step in the job. The routing data is used to determine the routing entry that identifies the program that is the process routing step.

QCMDB: The routing data used by the IBM-supplied batch subsystem to route batch jobs to the IBM-supplied control language processor, QCMD, is used.

***JOBD:** The routing data used to start the first routing step in the job description used with this job, is used.

***RQSDTA:** The request data (up to 80 characters), specified in the RQSDTA parameter of this command, is used.

'routing-data': Specify the character string that is used as the routing data for starting the first routing step. Up to 80 characters of text can be entered, enclosed in apostrophes if necessary.

RQSDTA

Specifies the request data that is placed as the last entry in this job's message queue. The request data can be a CL command run or a string of characters used by another program. For example, if RTGDTA(QCMDB) is specified, the IBM-supplied batch subsystem, QBATCH, is used, a CL command is supplied, and it becomes a message that is read by the control language processor, QCMD. Or, if a user program is specified in the routing entry, the request data can specify information, such as the record number of the first record in a file processed.

Note: If a value other than * is specified for this parameter, the data that follows the JOB command is ignored (it is not to be used as request data).

***:** The data following this BCHJOB command is inserted into this job's message queue as request data. For example, the request data may be a group of CL commands that constitute the job.

***JOBD:** The request data specified in the job description used by this job is placed as the last entry in this job's message queue.

***NONE:** No request data is placed in the job's message queue.

***RTGDTA:** The routing data in the RTGDTA parameter of this command is placed as the last entry in the job's message queue.

'request-data': Specify the character string that is placed as the last entry in the job's message queue. Up to of 256 characters can be entered, enclosed in apostrophes if necessary. If a CL command is entered, it must be enclosed in single apostrophes, and where apostrophes would normally be used inside the command, double apostrophes must be used.

SYNTAX

Specifies whether requests placed on the job's message queue are checked for syntax as CL commands. When checking for syntax is specified, the commands are checked for syntax when they are submitted instead of when the job is run, providing an earlier diagnosis of syntax errors. If checking is specified, the message

severity that causes a syntax error to end processing of a command is also specified. This parameter is used only if RQSDTA(*) is specified.

***JOBD:** The value in the job description used with this job determines whether the request data is checked for syntax and the message severity that is used.

***NOCHK:** The request data for this job is not checked for syntax as CL commands.

message-severity: Specify whether the request data is checked for syntax as CL commands, and, if a syntax error occurs that is equal to or greater than the error message severity specified, the running of the job that contains the command with errors is suppressed. Specify a value, ranging from 00 through 99, for the lowest message severity that causes running of the job to be suppressed. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

CURLIB

Specifies the name of the library being used as the current library during the processing of this command.

***USRPRF:** The current library in the user profile, under which this job is running, is used as the current library for this job.

***CRTDFT:** There is not a current library for the submitted job. If objects are created into the current library, the QGPL library is used as the default current library.

current-library-name: Specify the name of a library used as the current library for this job.

INLLIBL

Specifies the initial user library list that is used to search for any OS/400 system object names that were specified without a library qualifier.

Note: Duplicating libraries in the library list is not allowed. If specifying a list of libraries for INLLIBL (or if specifying *JOBD and the job description used specifies a list of libraries), ensure that there are no duplicates contained in the list being used. To do this, compare the libraries in the list to the libraries contained in system value QSYSLIBL. Duplicates must be removed from the INLLIBL parameter. If there is a current library associated with your user profile (CURLIB parameter on the CRTUSRPRF and CHGUSRPRF commands), that library cannot be used in the QSYSLIBL or QUSRLIBL system values, or on the list specified for INLLIBL.

***JOBD:** The library list in the job description used with this job is used as the initial user library list.

***SYSVAL:** The system default library list is used. It contains the library names that were specified in the system values, QSYSLIBL and QUSRLIBL, at the time that the job is started.

***NONE:** The user portion of the initial user library list is empty; only the system portion is used.

library-name: Specify the names of one or more libraries that are the user portion of the library list and are used by this job. No more than 25 names can be specified; the libraries are searched in the order in which they are listed.

ENDSEV

Specifies the message severity level of escape messages that can cause a batch job to end. The batch job is ended when a request in the batch input stream sends, to the request processing program, an escape message whose severity code is equal to or greater than that specified. (This type of end is considered an abnormal end.) This parameter value is compared with the severity of any unmonitored escape message that occurs as a result of running a noncompiled CL command in a batch job.

***JOBD:** The severity limit specified in the job description used with this batch job determines when the job is ended.

message-severity: Specify a value, ranging from 00 through 99, for the message severity of an escape message that results from a request in the batch input stream and that causes the job to end. Because escape messages sent to users can be up to a severity level of 50, a value of 50 or lower may be specified for a job being ended as a result of an escape message. An unhandled escape message, whose severity is equal to or greater than the value specified, causes the job to end. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

LOG

Specifies the message logging values used to determine the amount and type of information sent to the job log by this job. This parameter has three elements: the message (or logging) level, the message severity, and the level of message text. If no values are specified on this parameter, the values specified in the job description associated with this job are used.

Element 1: Message level

***JOBD:** The value specified for message logging in the job description is used.

message-level: Specify a value, ranging from 0 through 4, that specifies the message logging level used for this job's messages. For more information on the message levels, refer to the *message-level* variable under the CRTJOBD command's LOG parameter.

Element 2: Message Severity

***JOBD:** The value specified for message logging in the job description is used.

message-severity: Specify a value, ranging from 00 through 99, that specifies the lowest severity level to cause an error message to be logged in the job's log. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

Element 3: Message Text Level

***JOBD:** The value specified for message logging in the job description is used.

***MSG:** Only message text is written to the job's log or shown to the user.

***SECLVL:** Second-level message text is written to the job log.

***NOLIST:** No job log is produced if the job ends normally. If the job ends abnormally (if the end-of-job code is 20 or higher), a job log is produced. The messages appearing in the job log contain message text and help text.

LOGCLPGM

Specifies whether the commands that can be logged are logged to the job log through the CL program's message queue. This parameter sets the status of the job's logging flag. If *NO is specified, the logging flag status is off and CL commands are not logged. If *YES is specified and the LOG (*JOB) value has been specified in the CRTCLPGM command, all commands in the CL program that can be logged are logged to the job log.

For more information on request logging, refer to the LOG parameter in the CRTCLPGM command description.

***JOBD:** The value in the job description is used.

***NO:** The commands in a CL program are not logged to the job log.

***YES:** The commands in a CL program are logged to the job log.

INQMSGRPY

Specifies the way that predefined messages are answered (that is, predefined messages that are sent as a result of running this job). The user can specify that no change is made in the way that predefined messages are answered, that all inquiry messages require a reply, that a default reply be issued, or that the system reply list is checked for a matching reply as each predefined inquiry message is sent. Refer to the Add Reply List Entry (ADDRPYLE) command description for more information.

***JOBD:** The inquiry message reply control specified in the job description used with this job is started.

***RQD:** A reply is required by the receiver of the inquiry message for all inquiry messages that occur during the running of this job.

***DFT:** The default reply to the inquiry message is sent. If no default reply is specified in the message description of the inquiry message, the system default reply, *N, is used.

***SYSRPYL:** The system reply list is checked to see if there is an entry for any inquiry message issued as a result of running this job that has a message identifier and any comparison data that match the inquiry message identifier and message data. If a match occurs, the reply value in that entry is used. If no entry exists for that message, a reply is required.

PRTDEV

Specifies the qualified name of the default printer device for this job. If OUTQ(*DEV) is specified, the file is placed on an output queue with the same name as the printer.

***USRPRF:** The printer device name specified on the user profile of the user submitting this job is used.

***SYSVAL:** The value specified in the system value QPRTDEV is used.

***JOBD:** The printer device name specified in the job description is used.

printer-device-name: Specify the name of the printer device used.

OUTQ

Specifies the qualified name of the output queue used for spooled printer files that specify OUTQ(*JOB). This change does not affect files already created in active jobs or files in completed jobs in which the files were spooled.

***USRPRF:** The output queue specified on the user profile of the user submitting this job is used.

***DEV:** The output queue specified on the PRTDEV parameter is used.

***JOBD:** The output queue named in the job description used with this job is the default output queue.

The name of the queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

out-queue-name: Specify the name of the default output queue used by this job.

HOLD

Specifies whether jobs using this job description are placed on the job queue in the hold condition. A job placed on the job queue in the hold condition is held until it is either released by the Release Job (RLSJOB) command or canceled by the End Job (ENDJOB) or Clear Job Queue (CLRJOBQ) command. If the job is not run before the next power-down of the system, the job queue can be cleared (and the job ended) when the next initial program load (IPL) is done.

***JOBD:** The value specified in the job description determines whether this job is held when it is put on the job queue.

***NO:** The job is not held when it is put on the job queue.

***YES:** The spooled file is held until released by the Release Spool File (RLSSPLF) command.

DATE

Specifies the date that is assigned to the job when it is started.

***JOBD:** The date specified in the job description is used.

***SYSVAL:** The value in the QDATE system value at the time the job is started is used.

job-date: Specify the date when the job is started. The value must be entered using the system date format specified by the system value, QDATFMT.

SWS

Specifies the first settings for a group of eight job switches used with this job. These switches can be set or tested in a CL program and used to control the flow of the program. For example, if a certain switch is on, another program can be called. The job switches may also be valid in other high-level languages (HLL) programs. Only zeros (off) and ones (on) can be specified in the 8-digit character string.

***JOBD:** The value specified in the job description is the first setting for this job's switches.

switch-settings: Specify any combination of eight zeros and ones that is used as the first switch setting for this job.

MSGQ

Specifies the qualified name of the message queue to which messages are sent.

Note: If an abnormal ending occurs, the help text of the completion message that is sent specifies the possible causes.

***NONE:** No completion message is sent.

***USRPRF:** The message queue specified on the user profile of the user submitting this job is used.

The name of the queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

message-queue-name: Specify the name of the message queue where the completion message is sent.

SRTSEQ

Specifies the sort sequence table to be used for string comparisons for this job.

***USRPRF:** The sort table specified for the job's user profile is used. The user profile is specified in the job description, which is specified on the JOBD parameter.

***SYSVAL:** The system value QSRTSEQ is used.

***HEX:** A sort sequence table is not used. The hexadecimal values of the characters are used to determine the sort sequence.

***LANGIDUNQ:** A unique-weight sort table is used.

***LANGIDSHR:** A shared-weight sort table is used.

The name of the sort sequence table can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

table-name: Specify the name of the sort sequence table to be used with this job.

LANGID

Specifies the language identifier to be used when SRTSEQ(*LANGIDUNQ) or SRTSEQ(*LANGIDSHR) is specified.

***USRPRF:** The language ID specified for the job's user profile is used. The user profile is specified in the job description, which is specified on the JOBD parameter.

***SYSVAL:** The system value QLANGID is used.

language-ID: Specify the language identifier to be used by the job.

CNTRYID

Specifies the country identifier to be used by the job.

***USRPRF:** The country ID specified for the job's user profile is used. The user profile is specified in the job description, which is specified on the JOBD parameter.

***SYSVAL:** The system value QCNTRYID is used.

country-ID: Specify the country identifier to be used by the job.

CCSID

Specifies the coded character set identifier (CCSID) to be used for the job.

A CCSID is a 16-bit number identifying a specific set of encoding scheme identifiers, character set identifiers, code page identifiers, and additional coding-related information that uniquely identifies the coded graphic representation used.

***USRPRF:** The CCSID specified for the job's user profile is used. The user profile is specified in the job description, which is specified on the JOBD parameter.

***SYSVAL:** The CCSID specified for the QCCSID system value is used.

***HEX:** The CCSID 65535 is used.

coded-character-set-id: Specify the CCSID. More information on valid CCSIDs is in the *National Language Support Planning Guide*.

Examples

Example 1: Checking System Reply List for Inquiry Message Entries

```
BCHJOB  JOBD(PAYROLL)  INQMSGRPY(*SYSRPYL)
```

This command begins the batch job called PAYROLL. The job name is the same as the name of the job description used with the job. Except for the inquiry message control attribute, all of the values specified in the job description are used as the attributes of this job. An inquiry message that is sent (as a result of running this job) that has an entry in the system reply list is answered according to the reply in that reply list entry. For any inquiry message not represented in the reply list, a reply is required.

Example 2: Setting Job Switches

```
BCHJOB  JOBD(QBATCH)  JOB(PAYROLL)
  JOBQ(BATCH2)  INLLIBL(PAYLIB)
  SWS(00101100)  DATE(010188)
```

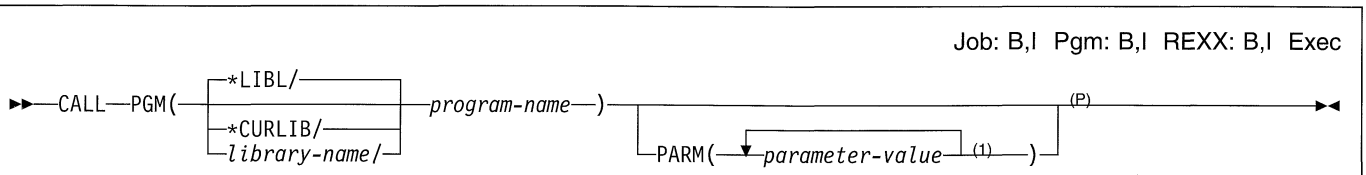
This command begins a batch job called PAYROLL, which is run using attributes from the IBM-supplied job description for batch jobs, QBATCH. The job is placed on the job queue BATCH2. The library PAYLIB is the only library in the user portion of the library list. Switches are set for use in the job, and the date is set at January 1, 1988.

Example 3: Specifying Severity Levels

```
BCHJOB  JOBD(COMPILE)  JOBPTY(5)
  SYNTAX(10)  INLLIBL(MYCMDS) ENDSEV(40)
```

This command begins a batch job called COMPILE, which is run using all of the attributes described in the job description also named COMPILE, except for the parameters that are changed by this command. The library MYCMDS is the only library in the user portion of the library list to be used when the commands are checked for syntax or run. Syntax errors with a value equal to or greater than 10 ends processing of the job. The job is assigned a scheduling priority of 5 and is run as long as no errors are encountered that cause an escape message to be sent that has a severity level of 40 or higher.

CALL (Call Program) Command



Notes:
1 A maximum of 40 repetitions
P All parameters preceding this point can be specified in positional form.

Purpose

The Call (CALL) command calls a program named on the command, and passes control to it. Optionally, the program or user issuing the CALL command can pass parameters to the called program. The CALL command can be used in batch jobs, in interactive jobs, and in both compiled and interpreted control language (CL). When the called program finishes processing, it can return control to the calling program using the RETURN command.

If the CALL command is issued by a CL program, each parameter value passed to the called program can be a character string constant, a numeric constant, a logical constant, a floating-point constant, or a CL program variable. If a floating-point constant is specified, the value is converted to double-precision format and passed to the called program. If parameters are passed, the value of the constant or variable is available to the program that is called. Parameters cannot be passed in any of the following forms: lists of values, qualified names, expressions, null parameters (that is, a parameter whose value is null, specified by *N), or keyword parameters. Up to 40 parameters can be passed to the called program.

If parameters are passed to a program using the CALL command, the values of the parameters are passed in the order in which they appear on the CALL command; this order must match the order in which they appear in the parameter list in the calling program.

Parameters in a called program can be used in place of its variables. However, no storage in the called program is associated with the variables it receives. Instead, if a variable is passed, the storage for the variable is in the program in which it was originally declared. If a constant is passed, a copy of the constant is made in the calling program and that copy is passed to the called program.

The result is that if a variable is passed, the called program can change its value and the change is reflected in the calling program. If a constant is passed, and its value is changed by the called program, the changed value is not known to the calling program. Therefore, if the calling program calls the same program again, the values of constants are set to their original values, but the variables do not change.

Information on passing variable parameters using the CALL command within a Submit Job (SBMJOB) command is in the *Work Management Guide*.

Restriction: The user must have object operational authority to the program being called. The user must also have a data authority.

Required Parameter

PGM

Specifies the qualified name of the program being called.

The name of the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the program being called.

Optional Parameter

PARM

Specifies one or more parameter values that are passed to the called program. Each of the values can be specified in only one of the following forms: a character string constant, a numeric constant, logical constant, double-precision floating point constant, or program variable.

The type and length of each parameter must be the same in both the calling and receiving programs. The number of parameters and the order in which they are sent and received must also be the same. If the CALL command is entered interactively or in a noncompiled batch environment, the type and length expected by the called program must match that of each parameter being passed on the command.

Parameters can be passed and received as follows:

- Character string constants of 32 bytes or less are *always* passed with a length of *32* bytes (padded on the right with blanks). If a character constant is longer than 32 bytes, the whole length of the constant is passed. If the parameter is defined to contain more than 32 bytes, the calling program must pass a constant that contains exactly that number of bytes. Constants longer than 32 characters are *not* padded to the length expected by the receiving program.

  The receiving program can receive less than the number of bytes passed (in this case, no message is sent). For example, if a program specifies that 4 characters are to be received and ABCDEF is passed (padded with blanks in 26 positions), only ABCD is accepted and used by the program. Quoted character strings can also be passed.
- Decimal constants are passed in packed form and with a length of (15 5), where the value is 15 digits long, of which 5 digits are decimal positions. If a parameter of 12345 is passed, the receiving program must declare the decimal field as (15 5); the parameter is received as 1234500000 (which is 12,345.00000).
- Logical constants are passed as 1 byte with a logical value of '1' or '0'.
- Floating-point literals and floating-point special values (*NAN, *INF, and *NEGINF) are passed as double-precision floating-point numbers, which occupy 8 bytes and are specified in the form ± n.n E ± n; for example, 2.47E3). A single-precision floating-point number cannot be passed to a called program.
- A program variable can be passed if the call is made from a CL program, in which case the receiving program must declare the field to match the variable defined in the calling CL program. For example, if a CL program defines a decimal variable named &CHKNUM as (5 0), the receiving program must declare the field as packed with 5 digits total, with no decimal positions.

  If either a decimal constant or a program variable can be passed to the called program, the parameter should be defined as (15 5), and any calling program must adhere to that definition. If the type, number, order, and length of the parameters do not match between the calling and receiving programs (other than the length exception noted previously for character constants), unpredictable results will occur.

  The value *N cannot be used to specify a null value because a null value cannot be passed to another program.

Examples

Example 1: Calling a Program

```
CALL   PGM(PAYROLL)
```

The program named PAYROLL is called with no parameters being passed to it. The library list is used to locate the called program.

Example 2: Defining a Character Constant

```
CALL   PAYROLL '1'
```

The program named PAYROLL is called with a character constant passed as a quoted string. The program must declare a field of up to 32 characters to receive the constant. The library list is used to locate the called program.

Example 3: Passing Parameters

```
CALL   LIB1/PAYROLL  (CHICAGO 1234 &VAR1)
```

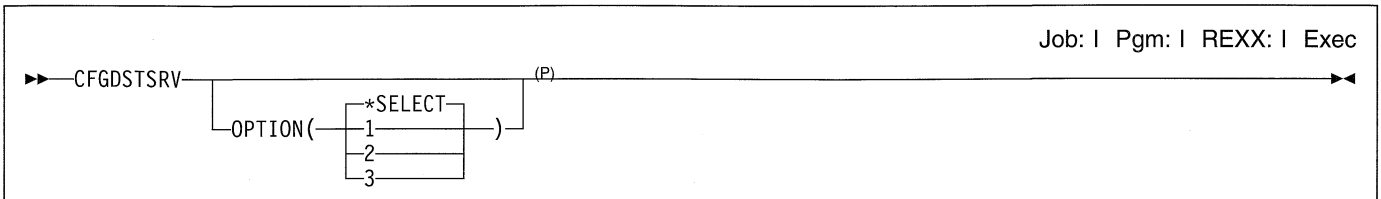
The program named PAYROLL located in library LIB1 is called. The calling program passes three parameters: a character string (CHICAGO), a decimal value (1234.00000), and the contents of the CL variable &VAR1. The attributes of the variable determine the attributes of the third parameter.

Example 4: Calling Program with Floating-Point Values

```
CALL   PGM1  (1.5E3 *INF)
```

The program named PGM1 is called with two double-precision floating-point values being passed to it.

CFGDSTSRV (Configure Distribution Services) Command



Note:

| P All parameters preceding this point can be specified in positional form.

Purpose

The Configure Distribution Services (CFGDSTSRV) command changes the configuration of the distribution network. The user can add, change, remove, and display entries from the distribution queues table, the routing table, and the secondary system name table. A detailed description of configuring a distribution network is in the *Distribution Services Network Guide*.

Restrictions:

1. This command is shipped with public *EXCLUDE authority and the QPGMR and QSYSOPR user profiles have private authorities to use the command.
2. Before this command is run for the first time, the QSNADS subsystem must be started to create the internal systems network architecture distribution services (SNADS) objects that this command uses.
3. Messages that report errors about system names or distribution queues may show or print different characters than the user entered because of internal system transformations. The internal value for a system name or distribution queue may differ from the characters shown by the CFGDSTSRV command depending on the language being used for the work station.

Optional Parameter

OPTION

Specifies an option from the Distribution Services Menu display that bypasses the initial menu and goes directly to the indicated table. The user can specify the distribution queues table, routing table, or secondary system name table without showing the Distribution Services Menu display.

***SELECT:** The option is selected from the Distribution Services Menu display.

1: The distribution queues table function, which identifies all the distribution queues for systems adjacent to the local system, is used.

An example of the distribution queues function is in the *Communications: Distribution Services Network Guide*.

2: The routing table function, which describes explicit or default entries for the destination systems in the SNADS network to which distribution queue entries can be routed, is used. An example of the routing table function is in the *Distribution Services Network Guide*.

3: The secondary system name function, which lists all names by which the system is known, is used. An example of the secondary system name function is in the *Distribution Services Network Guide*.

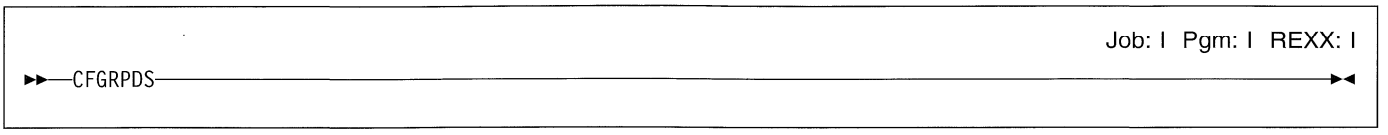
Example

```
CFGDSTSRV  OPTION(1)
```

This command shows the distribution queues' table entries.

Configuration changes may be made to existing distribution queues, or additional distribution queues may be configured.

CFGRPDS (Configure VM/MVS Bridge) Command



Purpose

The Configure Virtual Machine/Multiple Virtual Storage (VM/MVS) Bridge (CFGRPDS) command, previously named the Configure Remote Spooling Communications Subsystem/Professional Office System (RSCS/PROFS*) Distribution Services (CFGRPDS) command, configures the VM/MVS bridge application. The user can configure system network architecture distribution services (SNADS) and VM destinations, and can also enroll users in the system directory.

Restriction: This command is shipped with public *EXCLUDE authority and the QPGMR and QSYSOPR user profiles have private authorities to use the command.

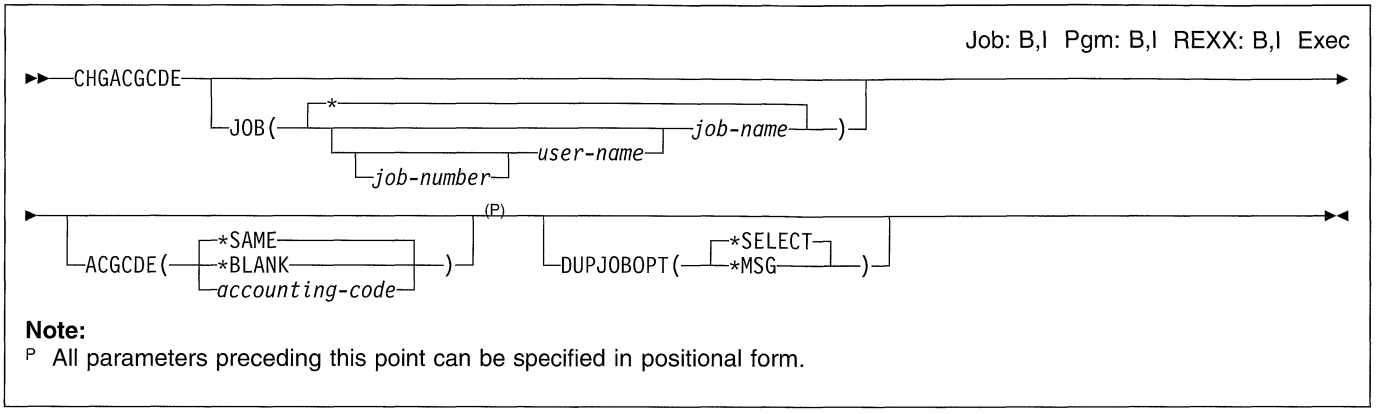
There are no parameters for this command.

Example

```
CFGRPDS
```

This command shows the Configure VM/MVS Bridge menu.

CHGACGCDE (Change Accounting Code) Command



Note:
P All parameters preceding this point can be specified in positional form.

Purpose

The Change Accounting Code (CHGACGCDE) command changes the accounting code of a job. The job can be on a job queue, or it can be active in a subsystem. This command has no effect if the job is on an output queue. If the command is entered when system value QACGLVL indicated that job accounting (*JOB) should be performed when the job entered the system, accounting information is journaled and a new accounting segment is started for the job. If the command is entered when the system value QACGLVL did not indicate job accounting should be performed, the accounting code is changed, but no journal entry is made. More information is in the *Work Management Guide*.

Restriction: To use this command, the user must be changing his own job, or he must have special job control authority.

Optional Parameters

JOB

Specifies the name of the job whose accounting code is changed.

A job identifier is a qualified name with up to three elements. For example:

job-name
user-name/job-name
job-number/user-name/job-name

More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***:** The job whose accounting code is changed is the job where this CHGACGCDE command is issued.

job-name: Specify the qualified name of the job whose accounting code is changed. If no job qualifier is given, all jobs currently in the system are searched for the job name. If more than one of the specified name is found, a qualified job name must be specified.

user-name: Specify the name of the user of the job whose accounting code is changed.

job-number: Specify the number of the job whose accounting code is changed.

ACGCDE

Specifies the accounting code used for the job.

***SAME:** The value does not change.

***BLANK:** The accounting code is changed to all blanks.

accounting-code: Specify the 15-character accounting code used for the next accounting segment. The accounting code may contain alphabetic or numeric characters. Blanks may also be used if the accounting code is enclosed in apostrophes.

DUPJOBOPT

Specifies the action taken when duplicate jobs are found by this command.

***SELECT:** The selection display is shown when duplicate jobs are found during an interactive session. Otherwise, a message is issued.

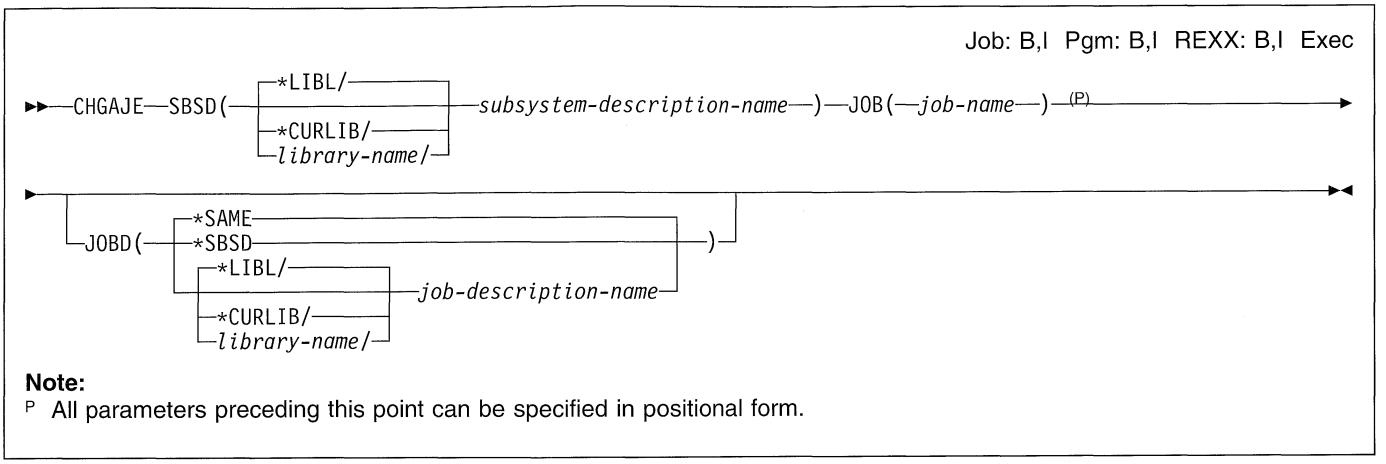
***MSG:** A message is issued when duplicate jobs are found.

Example

```
CHGACGCDE  JOB(123581/DEPT2/WS1)  ACGCDE(123456789)
```

This command changes the accounting code for job WS1, with user profile DEPT2, and job number 123581, to accounting code 123456789 for the next accounting segment. A job resource usage journal entry is written to the system accounting journal, QSYS/QACGJRN.

CHGAJE (Change Autostart Job Entry) Command



Note:
P All parameters preceding this point can be specified in positional form.

Purpose

The Change Autostart Job Entry (CHGAJE) command is used to specify a different job description for a previously defined job entry that starts automatically in the specified subsystem description.

Restriction: To use this command, the user must have object operational and object management authorities for the subsystem description and object operational authority for the job description.

Required Parameters

SBSD
Specifies the qualified name of the subsystem description containing the job entry being changed that starts automatically.

The name of the subsystem description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

subsystem-description-name: Specify the name of the subsystem description that contains the job entry.

JOB
Specifies the simple name that identifies the job entry that starts automatically in the subsystem description whose attributes are being changed.

Optional Parameter

JOBD
Specifies the qualified name of the job description used for the job that is started by this autostart job entry.

***SAME:** The value does not change.

***SBSD:** The job description having the same qualified name as the subsystem description, specified by the SBSD parameter, is used for the job that is started automatically.

The name of the job description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

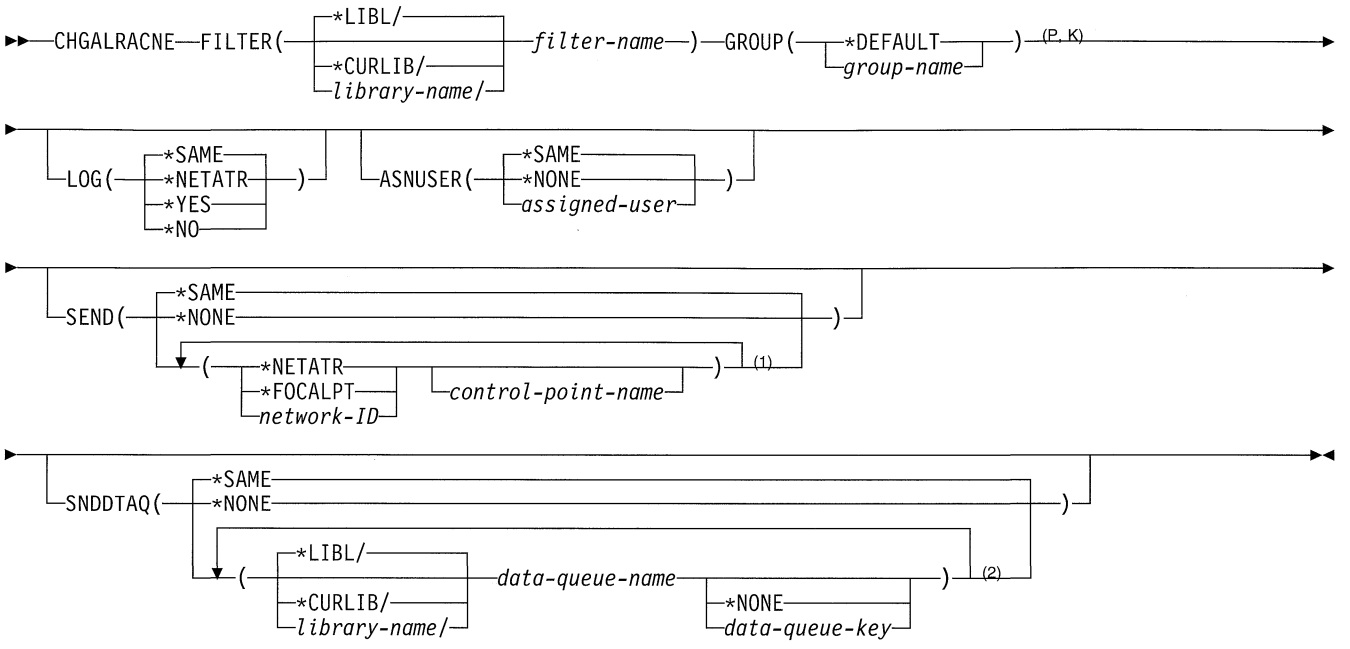
job-description-name: Specify the name of the job description used for the job being started by this job entry that starts automatically. If the job description does not exist when the entry is changed, a library qualifier must be specified because the qualified job description name is retained in the subsystem description.

Example

```
CHGAJE  SBSD(QGPL/PAYROLL)  JOB(INIT)  JOBD(MANAGER)
```

This command changes the JOBD parameter, for the job entry INIT that starts automatically, to MANAGER. The work entry is in the PAYROLL subsystem description in the QGPL library. The library list is used to locate the job description MANAGER. When the correct library is determined, the qualified job description name is placed in the subsystem description for this autostart job entry.

CHGALRACNE (Change Alert Action Entry) Command



Notes:

| P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

1 A maximum of 5 repititions

2 A maximum of 5 repititions

Purpose

The Change Alert Action Entry (CHGALRACNE) command allows the user to change an action entry in the specified alert filter. More information on alerts is in the *Alerts and DSNX Guide*.

Required Parameters

FILTER

Specifies the qualified name of the filter which contains the action entry being changed.

| The name of the filter can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

filter-name: Specify the name of the filter.

GROUP

Specifies the group of actions being changed.

***DEFAULT:** The default action entry that was automatically added when the filter was created is changed.

group-name: Specify the name of the group to which the defined actions are to be applied.

Optional Parameters

LOG

Specifies whether to log the alert.

***SAME:** The LOG action is not changed.

***NETATR:** The ALRLOGSTS network attribute controls the logging of this alert.

***YES:** The alert is logged.

***NO:** The alert is not logged.

ASNUSER

Specifies the user assigned to the alert.

***SAME:** The ASNUSER action is not changed.

***NONE:** No user is assigned to the alert.

assigned-user: Specify a user name.

SEND

Specifies the destination to which the alert is sent.

***SAME:** The destination does not change.

***NONE:** The alert is not sent.

Element 1: Network Identifier

***FOCALPT:** The alert is sent to the system focal point. The focal point system is determined at send time.

***NETATR:** The LCLNETID value specified in the system network attributes is used.

network-ID: Specify the network ID of the destination system.

Element 2: Control Point Name

control-point-name: Specify the control point name of the destination system.

SNDDTAQ

Specifies the data queue in which the alert notification record is placed. Keyed data queues are supported.

***SAME:** The data queue does not change.

***NONE:** No data queue is used.

The name of the data queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

Element 1: Data Queue Name

data-queue-name: Specify the name of the data queue.

Element 2: Data Queue Key

***NONE:** No key is used on the data queue.

data-queue-key: Specify the data queue key.

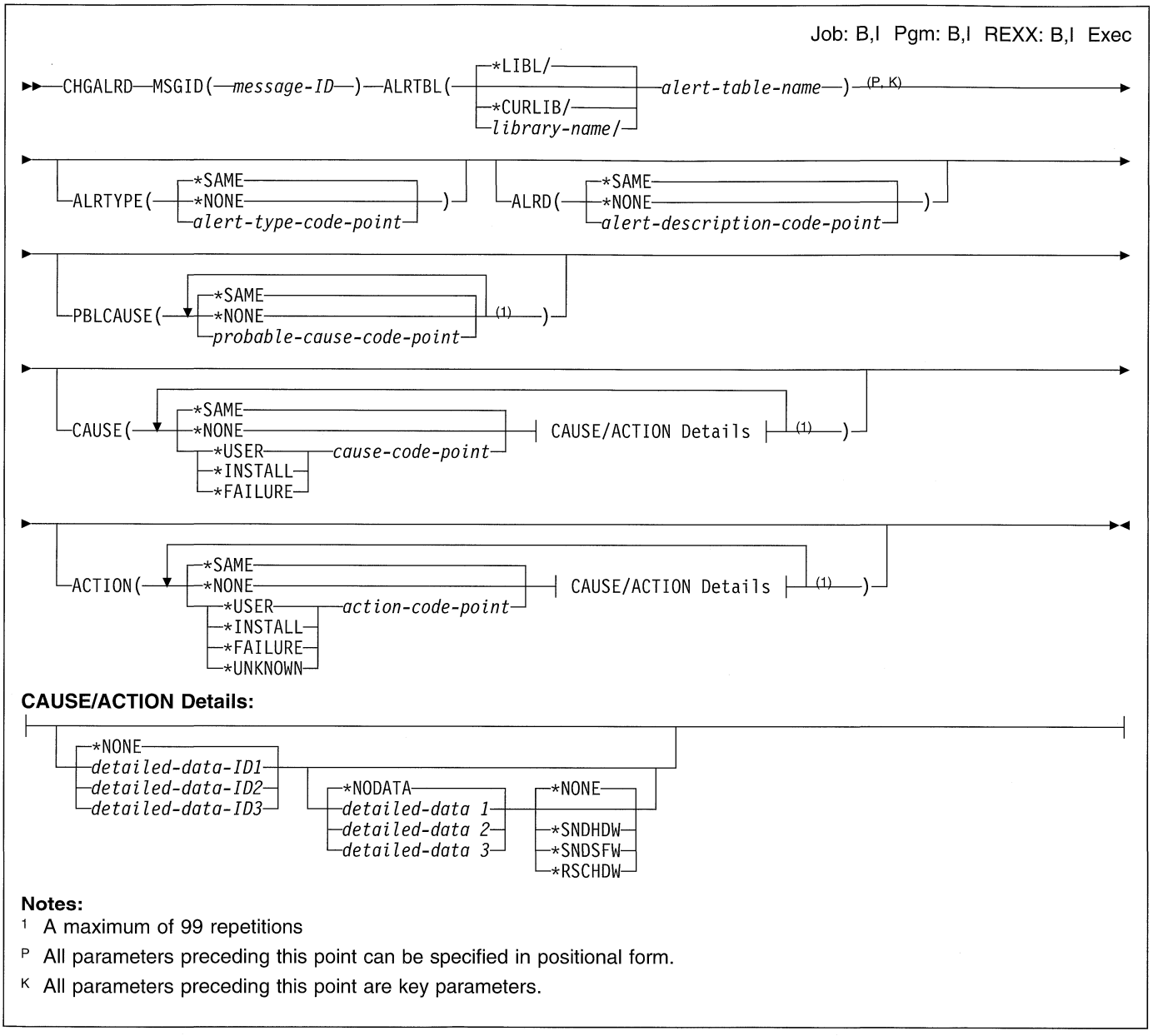
Example

```
CHGALRACNE FILTER(MYLIB/MYFILTER)  GROUP(CHICAGO)
  LOG(*SAME)  ASNUSER(CHICAGOOPR)
  SEND((*FOCALPT)(*NETATR.MILWKEE))  SNDDTAQ(*SAME)
```

This command changes actions for group CHICAGO to the following:

1. Use the same LOG action.
2. Send the alert to this system's focal point.
3. Send the alert to the system with control point name MILWKEE and a network id based on the LCLNETID value specified in the system network attributes.
4. Use the same SNDDTAQ action.
5. Assign the alert to user CHICAGOOPR.

CHGALRD (Change Alert Description) Command



Notes:

1 A maximum of 99 repetitions

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Alert Description (CHGALRD) command allows the user to change an alert description added previously by the add alert description command. More information on alerts is in the *Alerts and DSNX Guide*.

Required Parameters

MSGID

Specifies the message ID to which this alert description corresponds.

ALRTBL

Specifies the alert table in which this alert description is to be created.

The name of the alert table can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

alert-table-name: Specify the name of the alert table to be used.

Optional Parameters

ALRTYPE

Specifies the code point for the alert type.

***SAME:** The alert type code point does not change.

***NONE:** There is no alert type code point for this alert description.

alert-type-code-point: Specify the alert type code point to be used.

ALRD

Specifies the code point for the alert description.

***SAME:** The value does not change.

***NONE:** There is no alert description code point for this alert description.

alert-description-code-point: Specify the alert description code point to be used.

PBLCAUSE

Specifies up to 99 code points for probable causes, which are listed in order of decreasing probability.

***SAME:** The probable cause code point does not change.

***NONE:** There are no probable cause code points for this alert description.

probable-cause-code-point: Specify the probable cause code point to be listed.

CAUSE

Specifies user, install, or failure cause. Up to 99 causes can be listed.

***SAME:** The value does not change.

***NONE:** There are no cause code points for this alert description.

Element 1: Type of Cause Code Point

***USER:** A user cause code point follows.

***INSTALL:** An install cause code point follows.

***FAILURE:** A failure cause code point follows.

Element 2: Cause Code Point

cause-code-point: Specify the cause code point. Up to three detailed data qualifiers or one product identifier qualifier can be specified for each code point. A detailed data qualifier consists of a detailed data ID code point and detailed data. Specify *NONE *NODATA if there is no detailed data.

Element 3: First Detailed Data Identifier

***NONE:** There is no detailed data ID code point for this cause.

detailed-data-ID: Specify the detailed data identifier code point used to identify the data. Detailed data identifiers can be specified up to three times in each session.

Element 4: Detailed Data for First Identifier

***NODATA:** There is no data for this cause.

detailed-data: Specify up to 40 characters of detailed data. A substitution variable from the corresponding message description can be specified and the message data will be substituted into the alert description when the alert is created. Detailed data identifiers can be specified up to 3 times in each session.

Element 5: Product Identifier

***NONE:** There is no product identifier for this cause.

***SNDHDW:** Indicates the sender hardware (always AS/400).

***SNDSFW:** Indicates the sender software, specified in the PRDID keyword of the Create Alert Table (CRTALRTBL) command.

***RSCHDW:** Indicates the failing resource hardware, which is determined by the resource hierarchy in the message description.

Note: The user can enter either 0 to 3 detailed data identifiers or one product identifier, but not both.

ACTION

Specifies a recommended action for a user, install, or failure cause. Up to 99 actions can be listed.

***SAME:** The value does not change.

***NONE:** There are no action code points for this alert description.

Element 1: Type of Action Code Point

***USER:** A user cause code point follows.

***INSTALL:** An install cause recommended action code point follows.

***FAILURE:** A failure cause recommended action code point follows.

***UNKNOWN:** A recommended action for a 'cause undetermined' error follows.

Element 2: Action Code Point

action-code-point: Specify the recommended action code point. Up to three detailed data qualifiers or one product identifier qualifier can be specified for each code point. A detailed data qualifier consists of a detailed data ID code point and detailed data. Specify *NONE *NODATA if there is no detailed data.

Element 3: First Detailed Data Identifier

***NONE:** There is no detailed data ID code point for this action.

detailed-data-ID: Specify the detailed data identifier code point used to identify the data. Detailed data identifiers can be specified up to 3 times in each session.

Element 4: Detailed Data for First Identifier

***NODATA:** There is no data for this action.

detailed-data: Specify up to 40 characters of detailed data. A substitution variable from the corresponding message description can be specified and the message data will be substituted into the alert description when the alert is created. Detailed data identifiers can be specified up to 3 times in each session.

Element 5: Product Identifier

***NONE:** There is no product identifier for this action.

***SNDHDW:** Indicates the sender hardware (always AS/400).

***SNDSFW:** The sender software, specified in the PRDID keyword of the CRTALRTBL command.

***RSCHDW:** Indicates the failing resource hardware, which is determined by the resource hierarchy in the message description.

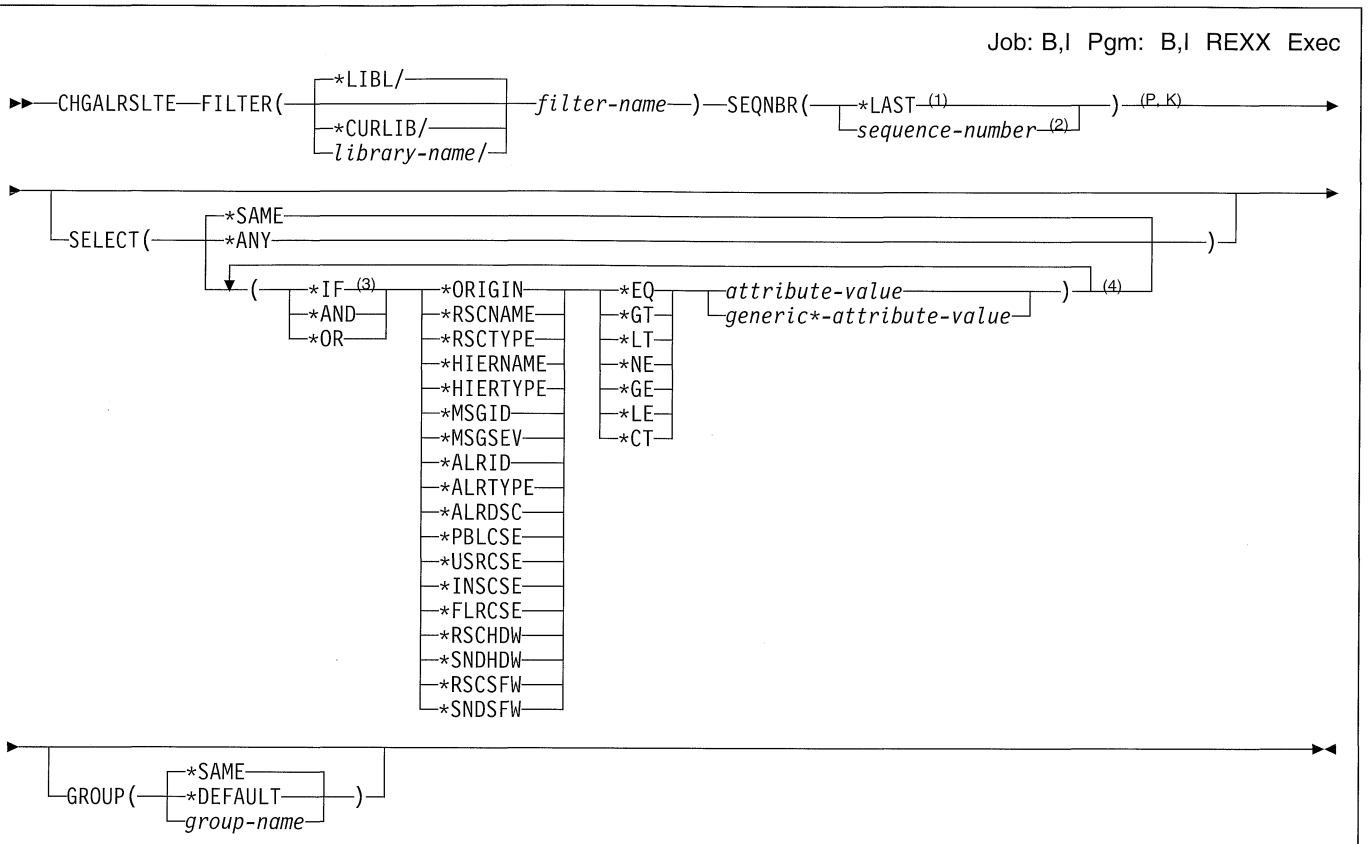
Note: The user can enter either 0 to 3 detailed data identifiers or one product identifier, but not both.

Example

```
CHGALRD   MSGID(USR1234)  ALRTBL(USER/USRMSGS)
  ALRTYPE(*SAME)  ALRD(*SAME)
  PBLCAUSE(1000 3121 6302)
  CAUSE(*SAME)  ACTION(*SAME)
```

This command adds probable cause 6302 to the alert description illustrated in the Add Alert Description (ADDALRD) command example.

CHGALRSLTE (Change Alert Selection Entry) Command



Notes:

1 If *LAST is specified then SELECT must be *ANY or *SAME.

2 Range is 1 through 9999

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

3 *IF must be specified as the first item only; either *AND or *OR must be specified when repeating selections.

4 A maximum of 10 repetitions

Purpose

The Change Alert Selection Entry (CHGALRSLTE) command allows the user to change an alert selection entry that was added previously using the Add Alert Selection Entry (ADDALRSLTE) command. More information on alerts is in the *Alerts and DSNX Guide*.

Required Parameters

FILTER

Specifies the qualified name of the filter in which the selection entry being changed.

The name of the filter can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

filter-name: Specify the name of the filter.

SEQNBR

Specifies the sequence number of the alert selection entry. Selection entries are evaluated in order by sequence number.

***LAST:** Allows the user to change the last selection. This entry is added automatically when the filter is created and will match any alert.

sequence-number: Specify a number from 1 through 9999.

Optional Parameters

SELECT

Specifies the comparisons to be made to determine if the alert belongs in the specified group. The selection entry results in a successful match with an alert when the data in the alert satisfies the relationships specified on the SELECT parameter. Up to 10 attribute values can be compared to the alert.

***SAME:** The value does not change.

***ANY:** Any alert matches this selection record. Specify the conditions under which an alert matches the selection entry. Each condition must contain the following four elements:

1. One of the logical operators *IF, *AND, or *OR
2. The attribute compared
3. One of the relational operators
4. The attribute value

Element 1: Logical Operator

***IF:** Identifies the first condition that must be satisfied.

***AND:** The conditions on both sides of the *AND must be satisfied.

***OR:** One of the conditions on each side of the *OR must be satisfied.

If there is one set or several sets of conditions, the *IF value must be specified as the first value in the first set of comparison values. If more than one set of conditions are specified, *AND or *OR must be specified as the first value in each set after the first. Each condition must be enclosed in parentheses. *AND is evaluated before *OR.

Element 2: Attribute

***ORIGIN:** Specifies whether the alert is generated or received. The valid values for this attribute are L (Locally generated) or R (Received).

***RSCNAME:** Specifies the name of the failing resource. The value for this attribute must be a 8-character name.

***RSCTYPE:** Specifies the type of the failing resource. The value for this attribute must be a 3-character resource type (for example, TAP or DKT).

***HIERNAME:** Specifies all of the resources in the alert resource hierarchy. The alert resource hierarchy is the list of resources, separated by blanks, displayed on the Work with Alerts (WRKALR) command detailed data displays. The value for this attribute can be a list of up to 5 resource names separated by a blank, unless the value is used with the *CT relational operator. If the *CT value is used, the selection relation can test to see if the given resource name is found anywhere within the hierarchy. This attribute contains the resource names from the hierarchy only.

***HIERTYPE:** Specifies all of the resource types in the alert resource hierarchy. The resource types match the resource names specified on the *HIERNAME attribute. The value for this attribute can be a list of up to 5 resource types (1 to 3 characters in length) separated by a blank, unless the value is used with the *CT relational operator. If the *CT value is used, the selection relation can test to see if the given resource type is found anywhere within the hierarchy.

***MSGID:** Specifies the message identifier.

***MSGSEV:** Specifies the message severity. This value must be greater than or equal to -2,147,483,647 and less than or equal to 2,147,483,647.

***ALRID:** Specifies the alert identifier. The alert identifier is displayed on the Work with Alerts (WRKALR) command detailed data display. The value for this attribute must be an 8-digit hexadecimal number unless it is used with the *CT relational operator. If the *CT operator or a wildcard character is used, the attribute must have an even number of digits up to a maximum of 8. The alert ID may not be a valid comparison for AS/400 alerts created after problem analysis.

***ALRTYPE:** Specifies the alert type code point that is in the alert. The value for this attribute is a 2 digit hexadecimal number.

***ALRDSC:** Specifies the alert description code point that is in the alert. The value for this attribute must be an 4-digit hexadecimal number unless it is used with the *CT relational operator. If the *CT operator or a wildcard character is used, the attribute must have an even number of digits up to a maximum of 4.

***PBLCSE:** Specifies the probable cause code point that is in the alert. The value for this attribute must be an 4-digit hexadecimal number unless it is used with the *CT relational operator. If the *CT operator or a wildcard character is used, the attribute must have an even number of digits up to a maximum of 4.

***USRCSE:** Specifies the first user cause code point that is in the alert. The value for this attribute must be an 4-digit hexadecimal number unless it is used with the *CT relational operator. If the *CT operator or a wildcard character is used, the attribute must have an even number of digits up to a maximum of 4.

***INSCSE:** Specifies the first install cause code point that is in the alert. The value for this attribute must be an 4-digit hexadecimal number unless it is used with the *CT relational operator. If the *CT operator or a wildcard character is used, the attribute must have an even number of digits up to a maximum of 4.

***FLRCSE:** Specifies the first failure cause code point that is in the alert. The value for this attribute must be an 4-digit hexadecimal number unless it is used with the *CT relational operator. If the *CT operator or a wildcard character is used, the attribute must have an even number of digits up to a maximum of 4.

***RSCHDW:** Specifies the failing hardware resource information in the alert. This information is displayed on the Work with Alerts (WRKALR) command detailed data

displays. Specify a value for this attribute using the following form:

```
'tttt mmm ss-ssssssss'
'tttt mmm ss-sssss'
'tttt mmm sssssss'
'tttt mmm sssss'
```

where tttt is the machine type, mmm is the model number, and sssssssss is the serial number. Use this format to match a particular hardware resource or use a part of the hardware value with the *CT relational operator to provide a partial match.

***SNDHDW:** Specifies the sending hardware resource information in the alert. This information is displayed on the Work with Alerts (WRKALR) command detailed data displays. Specify a value for this attribute using the following form:

```
'tttt mmm ss-ssssssss'
'tttt mmm ss-sssss'
'tttt mmm sssssss'
'tttt mmm sssss'
```

where tttt is the machine type, mmm is the model number, and sssssssss is the serial number. Use this format to match a particular hardware resource or use a part of the hardware value with the *CT relational operator to provide a partial match.

***RSCSFW:** Specifies the failing software resource information in the alert. This information is displayed on the Work with Alerts (WRKALR) command detailed data displays. Specify a value for this attribute using the following form:

```
'ppppppp vv rr mm'
```

where ppppppp is the licensed program identifier, vv is the version number, rr is the release number, and mm is the modification level. Use this format to match a particular software resource or use a part of the software value with the *CT relational operator to provide a partial match.

***SNDSFW:** Specifies the sending software resource information in the alert. This information is displayed on the Work with Alerts (WRKALR) command detailed data displays. Specify a value for this attribute using the following form:

```
'ppppppp vv rr mm'
```

where ppppppp is the licensed program identifier, vv is the version number, rr is the release number, and mm is the modification level. Use this format to match a particular software resource or use a part of the software value with the *CT relational operator to provide a partial match.

Element 3: Relational Operator

***EQ:** The attribute in element 2 must be equal to the value specified in element 4.

***GT:** The attribute in element 2 must be greater than the value specified in element 4.

***LT:** The attribute in element 2 must be less than the value specified in element 4.

***NE:** The attribute in element 2 must not be equal to the value specified in element 4.

***GE:** The attribute in element 2 must be greater than or equal to the value specified in element 4.

***LE:** The attribute in element 2 must be less than or equal to the value specified in element 4.

***CT:** The attribute in element 2 must contain the value specified in element 4.

Element 4: Attribute Value

attribute-value: Specify the value (a maximum of 60 characters) to be compared with the contents of the specified attribute. The value must be specified in apostrophes if it contains blanks or special characters and must be in character format. If a CL variable is specified for the value, it must be a character variable.

generic-attribute-value:* Specify the generic attribute value. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all attributes with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete attribute name. If the complete attribute name is specified, and multiple libraries are searched, multiple attributes can be changed only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

GROUP

Specifies the group that an alert is assigned to if the alert matches the criteria specified on the SELECT parameter.

***SAME:** The value does not change.

***DEFAULT:** The alert is assigned to the *DEFAULT group. The *DEFAULT group is automatically added when a filter is created.

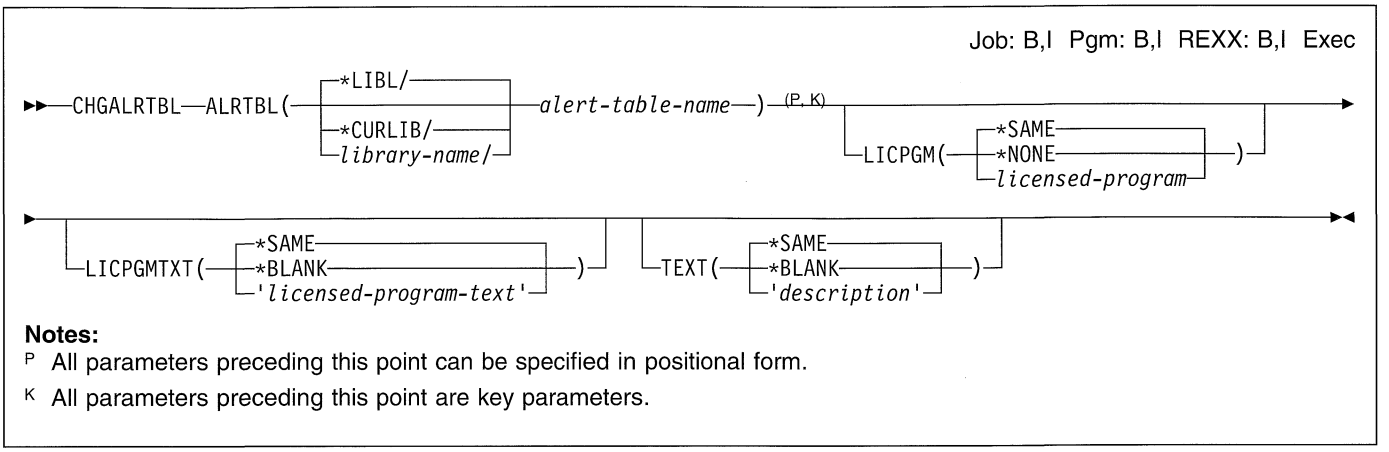
group-name: Specify a group name to which the alert is assigned.

Example

```
CHGALRSLTE  FILTER(MYLIB/MYFILTER)  SEQNBR(10)
  SELECT(*SAME)  GROUP(NEWSTUFF)
```

This command changes the GROUP to NEWSTUFF for selection entry 10 in the filter MYFILTER in library MYLIB.

CHGALRTBL (Change Alert Table) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Alert Table (CHGALRTBL) command is used to change one of the values defined by the Create Alert Table (CRTALRTBL) command. Alert tables define alerts, which are problem notifications in a network. The CHGALRTBL command can be used to change the product identification, product text, or object text for an alert table. The typical user of the CHGALRTBL command is the system or network programmer or operator responsible for network management. More information on alerts is in the *Alerts and DSNX Guide*.

Required Parameter

ALRTBL

Specifies the name of the alert table in which the alert description is changed.

The name of the alert table can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

alert-table-name: Specify the name of the alert table.

Optional Parameters

LICPGM

Specifies the licensed program that the alert table is identifying. This program is included in the alert as product identification for the alert sender.

***SAME:** The value does not change.

***NONE:** There is no licensed program for this alert table. This value is allowed for products that do not have a licensed program.

licensed-program: Specify a 7-character identifier for the licensed program. This identifier retrieves release and level information for the program.

Note: The licensed program is not necessarily an IBM Licensed Program. Any 7-character identifier that is meaningful to the use of the alerts can be specified (for example, USR1234). If the value is assigned to the system, the identifier and release level information are included in the alert. If the value is not known, then only the identifier and text specified in LICPGMTXT are included in the alert.

LICPGMTXT

Specifies descriptive text for the alert table product identifier (for example, OS/400). This text is included in the alert as product identification for the alert sender.

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'licensed-program-text': Specify up to 30 characters of text, enclosed in apostrophes, describing the licensed program text.

TEXT

Specifies text that briefly describes the alert table. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

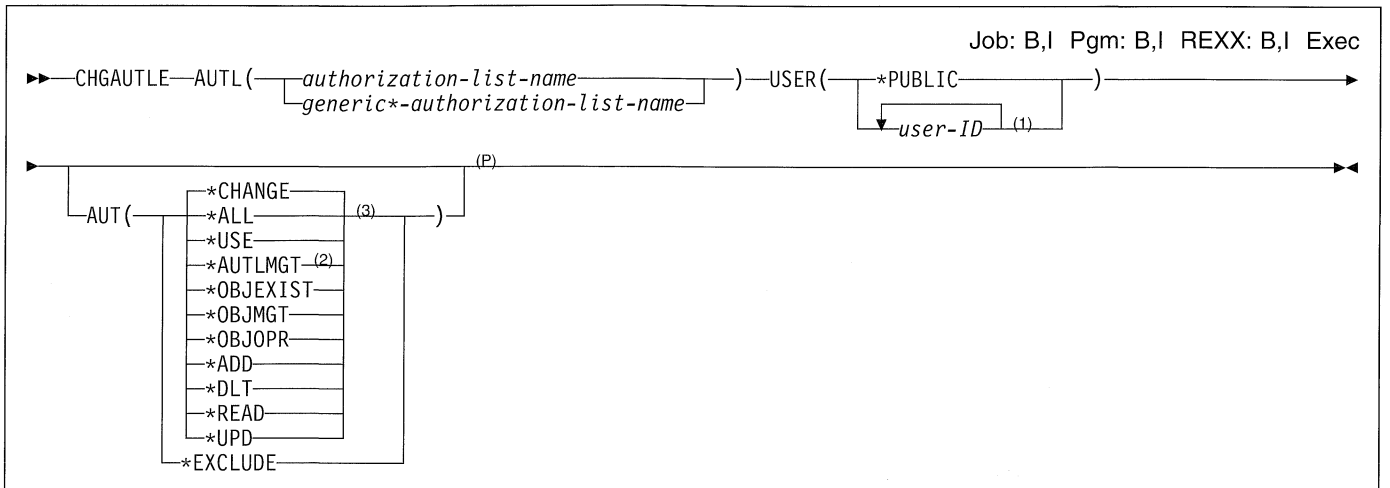
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGALRTBL  ALRTBL(ALRTBLLIB/ALRTBLNBR1)
  LICPGMTXT('OS/400--customer defined')
```

This command changes the licensed program for the alert table in library ALRTBLLIB called ALRTBLNBR1.

CHGAUTLE (Change Authorization List Entry) Command



Notes:

1 A maximum of 50 repetitions

2 *AUTLMGT is not valid if USER(*PUBLIC) is specified.

3 Select one or more, 8 maximum

P All parameters preceding this point can be specified in positional form.

Purpose

The Change Authorization List Entry (CHGAUTLE) command changes the authorities for users on authorization lists. The authorities that the users have on the authorization list are replaced with the authorities specified on the command. The authorization list must already exist and the users must be on the list. If the user specified is not on the list, a message is issued.

The users who can use this command to change the authorization list are: the owner of the authorization list, a user with authorization list management authorities on the list, or a user with all object authority.

When the CHGAUTLE command is used to change a user's authorities, the user must specify the name of the authorization list, a list of users, and a list of authorities. All users specified in the list are given the same authorities. The authorities of each user on the list given to the command are changed to the authorities specified on the command. Authority can be specified for all users who do not have specific authority, who are not on the authorization list, and whose group has no authority, by giving a user ID of *PUBLIC.

Restrictions:

1. Only the owner of the list or a user with *ALLOBJ authority can change a user's authorities to include *AUTLMGT.
2. A user with *AUTLMGT authority can change a user's authority only to the *AUTLMGT level. If the change results in adding or removing a user's authority, the current user of this command must also have the specific authority being added or removed.

Required Parameters

AUTL

Specifies the name or generic name of the authorization list for which users' authorities are to be changed. The authorization list must already exist.

authorization-list-name: Specify the name of the authorization list used.

generic-authorization-list-name:* Specifies the generic name of the authorization list. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be changed only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

USER

Specifies a list of user IDs whose authorities on the authorization list are changed. Up to 50 user IDs can be specified. If a user ID is not on the authorization list, a message is issued.

***PUBLIC:** Authority is given to all users who have no specific authority, are not on the authorization list, and whose group does not have any authority.

user-ID: Specify a list of user IDs whose authorities are changed.

Optional Parameter

AUT

Specifies the authority given to users specified on the USER parameter. Users must have *AUTLMGT authority to manage the authorization list.

***CHANGE:** The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority. If the alert table is an authorization list, the user cannot add, change, or remove user IDs.

***ALL:** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence and specify the security for the object, change the object, and perform basic functions on the object. The user can change ownership of the object.

***USE:** The user can perform basic operations on the object, such as running a program or reading a file. The user cannot change the object. *USE authority provides object operational authority and read authority.

***EXCLUDE:** The user cannot access the object.

***AUTLMGT:** Authorization list management authority provides the authority to add users to the authorization list, change users' authorities on the authorization list, or remove users from the authorization list. This parameter is not valid when USER(*PUBLIC) is specified.

***OBJEXIST:** Object existence authority provides the authority to control the object's existence and ownership. These authorities are necessary for users who want to delete the object, free storage for the object, perform save and restore operations for the object, or transfer ownership of the object. A user with special save system authority (*SAVSYS) does not need object existence authority. Object existence authority is required to create an object that was named by an authority holder.

***OBJMGT:** Object management authority provides the authority to specify the security for the object, move or rename the object, and add members to database files.

***OBJOPR:** Object operational authority provides authority to look at the description of the object and to use the object as determined by the data authority that the user has to the object.

***ADD:** Gives the authority to add entries to an object (for example, job entries to a queue or records to a file).

***DLT:** Delete authority allows the user to remove entries from an object, for example, remove messages from a message queue or records from a file.

***READ:** Read authority provides the authority needed to show the contents of an entry in the object or to run a program.

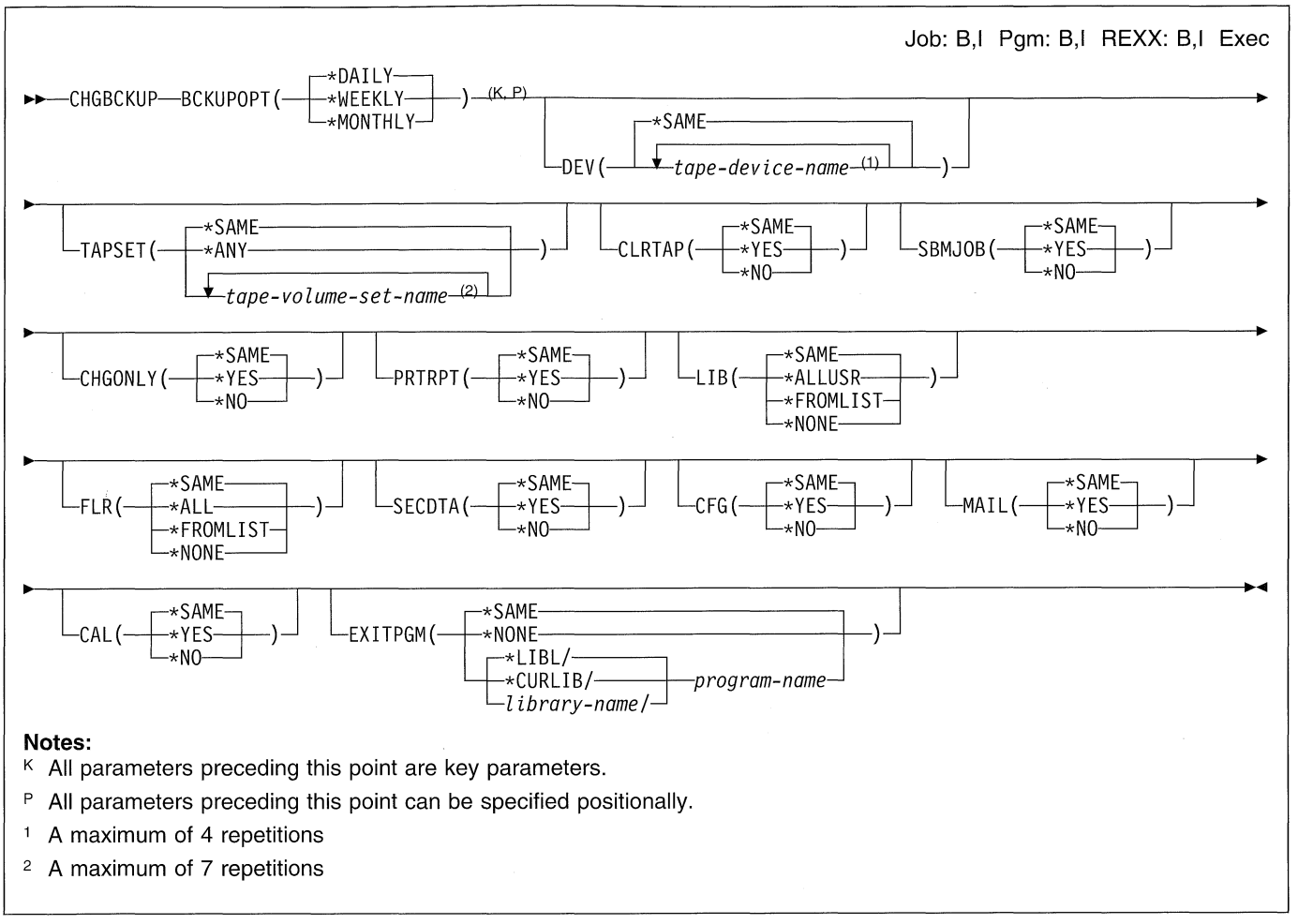
***UPD:** Update authority provides the authority needed to change the entries in the object.

Example

```
CHGAUTLE  AUTL(DEPT48X)
  USER(KARENG KARENS JEFF JULIE DARL)
  AUT(*CHANGE)
```

This command changes the authority that users KARENG, KARENS, JEFF, JULIE, and DARL have on the authorization list to *CHANGE. *CHANGE gives the users object operational authority and all data authorities to the objects secured by the authorization list.

CHGBCKUP (Change Backup Options) Command



Notes:

K All parameters preceding this point are key parameters.

P All parameters preceding this point can be specified positionally.

1 A maximum of 4 repetitions

2 A maximum of 7 repetitions

Purpose

The Change Backup Options (CHGBCKUP) command allows the user to change the options in one of the predefined backups. More information on backup is in the *Advanced Backup and Recovery Guide*.

Required Parameter

BCKUPOPT

Specifies the backup options to be changed.

***DAILY:** The options for the daily backup are changed.

***WEEKLY:** The options for the weekly backup are changed.

***MONTHLY:** The options for the monthly backup are changed.

Optional Parameters

DEV

Specifies the tape device to use for the backup.

***SAME:** The tape device name stored in the specified options is not changed.

tape-device-name: Specify a list of tape device names to use with the specified backup options.

TAPSET

Specifies the name of the tape set to be used.

***SAME:** The tape set name stored in the specified options is not changed.

***ANY:** The tapes mounted on the backup devices are used for the backup. Tape volume IDs are not checked.

tape-volume-set-name: Specify a list of 4-character names of tape volume sets to be rotated for the backup. The tape volume IDs for the backup are generated by concatenating sequential numbers starting with '01' to the specified prefix.

CLRTAP

Specifies whether to clear the tape and start the save at sequence number 1.

***SAME:** The Clear Tape indicator stored in the specified options is not changed.

***YES:** The tape is cleared and the save starts at sequence number 1 (equivalent to CLEAR(*ALL) SEQNBR(1) on the SAVxxx commands).

***NO:** The tape is not cleared and the save starts after the last active file on the tape (equivalent to CLEAR(*NONE) SEQNBR(*END) on the SAVxxx commands).

SBMJOB

Specifies whether to submit the backup as a batch job when the RUNBCKUP menu is used to run a backup using these options.

Note: This parameter is ignored when the RUNBCKUP command is used to run a backup.

***SAME:** The Submit Job indicator stored in the specified options is not changed.

***YES:** The backup is submitted as a batch job when the menu is used to perform the backup.

***NO:** The backup is run interactively when the menu is used to perform the backup.

CHGONLY

Specifies whether to save only changed objects in the libraries and folders being backed up.

***SAME:** The Save Changed Only indicator stored in the specified options is not changed.

***YES:** Only objects changed since the last backup are saved.

***NO:** All of the objects in the requested libraries and folders are backed up.

PRTRPT

Specifies whether a detailed list of saved objects is printed. A summary report is always printed.

***SAME:** The Print Report indicator stored in the specified options is not changed.

***YES:** A detailed list of saved objects and a summary report are printed.

***NO:** A summary report is printed.

LIB

Specifies which libraries are backed up.

***SAME:** The libraries specified in the options are not changed.

***ALLUSR:** Performs a backup of all user libraries. All user libraries are backed up. All libraries with names that do not begin with the letter Q are backed up except for the following:

```
#CGULIB     #DFULIB     #RPGLIB     #SEULIB
COBLIB      #DSULIB     #SDALIB
```

Although the following Qxxx libraries are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are considered *user libraries*, and are also backed up:

```
QDSNX       QPFRDATA    QUSER38
QGPL        QRCL        QUSRSYS
QGPL38      QS36F       QUSRVxRxMx
```

Note: A different library name, of the form QUSRVxRxMx, is added with each release. VxRxMx is the version, release, and modification level of the library.

***FROMLIST:** The libraries selected for backup in the library backup list are backed up.

***NONE:** No libraries are backed up.

FLR

Specifies which folders are backed up.

***SAME:** The folders specified in the options are not changed.

***ALL:** All folders are backed up.

***FROMLIST:** The folders selected for backup in the folder backup list are backed up.

***NONE:** No folders are backed up.

SECDTA

Specifies whether to save the system security data.

***SAME:** The Security Data indicator stored in the specified options is not changed.

***YES:** Security data is saved when this backup is run.

***NO:** Security data is not saved.

CFG

Specifies whether to save the system configuration data.

***SAME:** The Configuration Data indicator stored in the specified options is not changed.

***YES:** Configuration data is saved when this backup is run.

***NO:** Configuration data is not saved.

MAIL

Specifies whether to save OfficeVision/400* mail. This parameter is ignored if FLR(*ALL) is specified.

***SAME:** The Mail indicator stored in the specified options is not changed.

***YES:** Mail is saved when this backup is run.

***NO:** Mail is not saved.

CAL

Specifies whether to save OfficeVision/400 calendar data. OfficeVision/400 calendars are also saved when QUSRSYS is saved.

***SAME:** The calendar indicator stored in the specified options is not changed.

***YES:** Calendars are saved when this backup is run.

***NO:** Calendars are not saved.

EXITPGM
Specifies the user program to call before the backup begins and again after the backup is complete.

***SAME:** The program name stored in the specified options is not changed.

***NONE:** No exit program is called.

The name of the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the program to call before and after the backup.

Examples

Example 1: Changing the Daily Backup Options

```
CHGBCKUP  BCKUPOPT(*DAILY)  MAIL(*YES)  CAL(*YES)
```

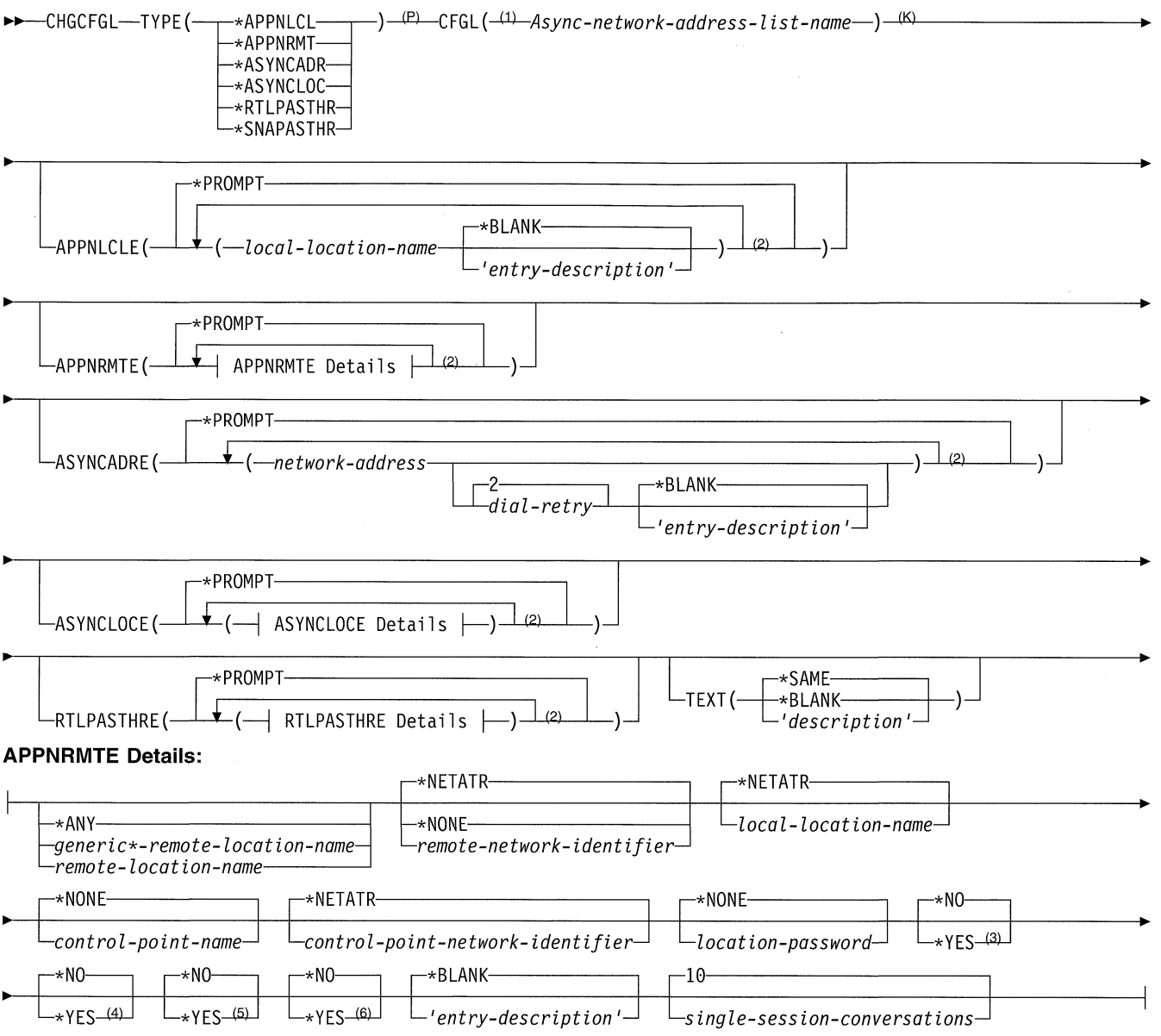
This command changes the daily backup to save the OfficeVision/400 mail and calendar data.

Example 2: Changing the Monthly Backup Options

```
CHGBCKUP  BCKUPOPT(*MONTHLY)  DEV(TAP01 TAP02)
TAPSET(RED GRN BLU)
```

This command changes the monthly backup to use tape devices TAP01 and TAP02 and tape sets RED, GRN, and BLU.

CHGCFGL (Change Configuration List) Command



Notes:

1 Required only for TYPE(*ASYNCADR)

2 A maximum of 50 repetitions

3 Location Security

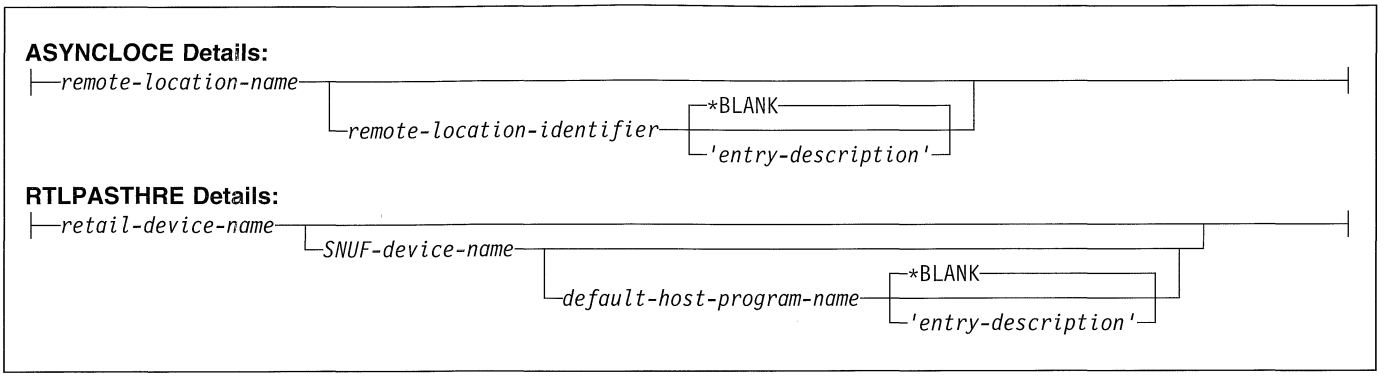
4 Single Session Location

5 Locally Controlled Session

6 Preestablished Session

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.



Purpose

The Change Configuration List (CHGCFGL) command changes a configuration list.

Required Parameters

TYPE

Specifies the type of the configuration list to be changed.

***APPNLCL:** An APPN local configuration list is used. Up to 476 APPN local location entries are allowed in the configuration list.

***APPNRMT:** An APPN remote configuration list is used. Up to 1898 APPN remote location entries are allowed in the configuration list.

***ASYNCADR:** An asynchronous network address configuration list is used. Up to 294 asynchronous network address entries are allowed in the configuration list.

***ASYNCLOC:** An asynchronous remote location configuration list is used. Up to 4995 asynchronous remote location entries are allowed in the configuration list.

***RTLPASTHR:** A retail pass-through list is used. Up to 450 retail pass-through entries can be specified in the configuration list.

***SNAPASTHR:** An SNA pass-through list is used. Only the text description for an SNA configuration list can be changed using the Change Configuration List (CHGCFGL) command.

Note: To change an entry in an SNA configuration list, use the Change Configuration List Entry (CHGCFGLE) command.

CFGL

Specifies the name of the configuration list. This parameter is valid only when *ASYNCADR is specified on the TYPE parameter. Only one of the other configuration list types is allowed on a system. The list types have system-supplied names: QAPPNLCL, QAPPNRMT, QASYNCADR, QASYNCLOC, QRTLPASTHR, QSNAPASTHR.

Optional Parameters

APPNLCLE

Specifies the APPN local location entry. This value is required if *APPNLCL is specified for the TYPE parameter.

***PROMPT:** The value of *PROMPT allows the user to add, remove, and change entries using a full-screen entry display.

Element 1: Operation on a Local Entry

local-location-name: Specify the location name residing on the local system. This name is used by APPN to determine whether the request being received is for this system or for another system in the network. The local location name must be unique and cannot already exist as a remote location name used by configuration list QAPPNRMT or QAPPNLCL on a different system in the APPN network.

Element 2: Describing a Local Entry

***BLANK:** Text is not specified.

'entry-description': Specify a short description of 20 characters or less for each local entry.

APPNRMTE

Specifies the APPN remote location entry. This value is required if *APPNRMT is specified for the TYPE parameter.

***PROMPT:** The value of *PROMPT allows the user to add, remove, and change entries using a full-screen entry display.

Element 1: Remote Location Name

***ANY:** The system potentially accepts all requests sent to it.

generic-remote-location-name:* Specify the generic name (part of a name followed by an asterisk) of the remote location(s) to be changed. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the com-

plete object name is specified, and multiple libraries are searched, multiple objects can be changed only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

remote-location-name: Specify the full name of a remote location.

Element 2: Remote Network Identifier

***NETATR:** The RMTNETID value specified in the system network attributes is used.

***NONE:** No remote network identifier (ID) is used.

remote-network-identifier: Specify the network identifier of the network in which the remote location resides.

Element 3: Local Location Name

***NETATR:** The LCLLOCNAME value specified in the system network attributes is used.

local-location-name: Specify the location name residing on the local system. This name is used by APPN to match a local/remote location pair entry.

Element 4: Control Point Name

***NONE:** No control point name is specified.

control-point-name: Specify the control point providing network functions for the remote location. This field is required if the entry is generic or *ANY.

Element 5: Control Point Network Identifier

***NETATR:** The LCLNETID value specified in the system network attributes is used.

control-point-network-identifier: Specify the network identifier of the network in which the control point resides.

Element 6: Location Password

***NONE:** No location password is specified.

location-password: Specify the password that is used when establishing sessions on the local location/remote location name pair. It must be an even number of hexadecimal characters.

Element 7: Location Security

This value specifies whether the local location allows the remote location to verify user passwords when sending program start requests to the local location.

***NO:** The remote location does not verify user passwords when sending program start requests to the local location.

***YES:** The remote location can verify user passwords when sending program start requests to the local location.

Element 8: Single Session Location

This value specifies whether the connection between the local location and remote location is a single session connection.

***NO:** A single session connection is not made between the local and the remote location.

***YES:** A single session connection is made between the local location and the remote location.

Element 9: Locally Controlled Session

This value specifies whether the single session connection between the local location and remote location is locally controlled.

***NO:** The single session connection does not have to be locally controlled.

***YES:** The single session connection is locally controlled.

Element 10: Preestablished Session

This value specifies whether the single session is automatically bound when the mode is started between the local location and remote location.

***NO:** The single session connection is not automatically made between the local location and the remote location.

***YES:** The single session connection is automatically made between the local location and the remote location.

Element 11: Remote Entry Description

***BLANK:** Text is not specified.

'entry-description': Specify no more than 20 characters of text, enclosed in apostrophes for the remote entry description.

Element 12: Number of Single-Session Conversations

10: The number of single session conversations allowed is ten.

single-session-conversations: Specify the number of conversations allowed for a single session. Valid values range from 1 through 512.

Note: The combination of remote location name, remote network identifier, and local location name must be unique. Also, the remote location name cannot already exist as a local location in configuration list QAPPNLCL, or as the current value for LCLLOCNAME or CPNAME network attribute.

ASYNCADRE

Specifies the asynchronous network address entry. This value is required if *ASYNCADR is specified for the TYPE parameter.

***PROMPT:** The value of *PROMPT allows the user to add, remove, and change entries using a full-screen entry display.

Element 1: Network Address

network-address: Specify the X.25 network address. Valid values range from 0 through 9.

Element 2: Number of Dial Retries

2: The default number of retries is two.

dial-retry: Specifies the number of times that dialing is retried after errors occur before attempting to dial the next number on the list. Valid values range from 1 through 255.

Element 3: Description of Network Address Entry

***BLANK:** Text is not specified.

'entry-description': Specify no more than 20 characters of text, enclosed in apostrophes for the description of the network address entry.

ASYNCLOCE

Specifies the asynchronous remote location entry. This value is required if *ASYNCLOC is specified for the TYPE parameter.

***PROMPT:** The value of *PROMPT allows the user to add, remove, and change entries using a full-screen entry display.

Element 1: Remote Location Name

remote-location-name: Specify the name that, when combined with the remote location identifier, determines whether to accept an incoming call. It is the same name used in the remote system as its local name. This value must be unique.

Element 2: Remote Location Identifier

remote-location-identifier: Specify an identifier that, when combined with the remote location name, determines if an incoming call is accepted. This identifier must be the same as the local identifier of the remote system.

Element 3: Description of Remote Location Entry

***BLANK:** Text is not specified.

'entry-description': Specify a short description of 20 characters or less for each remote location entry.

RTLPASTHRE

Specifies the retail pass-through entry. This value is required if TYPE(*RTLPASTHR) is specified. Up to 50 entries can be specified for this parameter.

***PROMPT:** The value of *PROMPT allows the user to add, remove, and change entries using a full-screen entry display.

Element 1: Retail Device Name

retail-device-name: Specify the name of the retail device to use for the pass-through session. This must be a unique value.

Element 2: SNA Upline Facility Device Name

SNUF-device-name: Specify the name of the host device to use for the pass-through session. This must be a unique value.

Element 3: Default Host Program Name

default-host-program-name: Specify the name of the program to be started on the host if a program name was not specified by the retail controller.

Element 4: Text Description

***BLANK:** Text is not specified.

'entry-description': Specify a short description of 20 characters or less for each retail pass-through entry.

TEXT

Specifies text that briefly describes the object. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

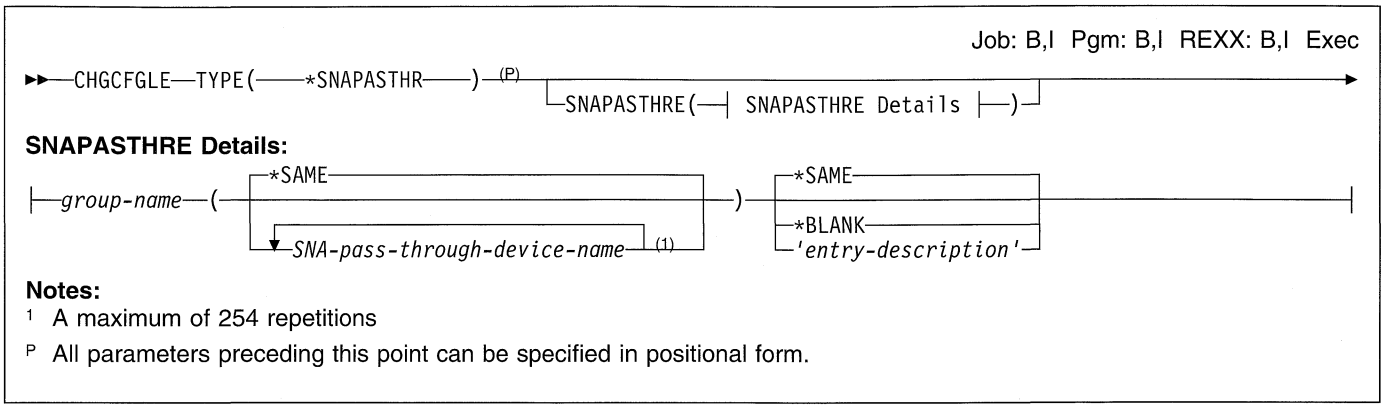
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGCFGL  TYPE(*ASYNCADR)  CFGL(CFGL01)
```

This command brings up a full-screen entry display, showing existing entries for configuration list CFGL01. From the entry display, the user can add, change, and remove entries.

CHGCFGLE (Change Configuration List Entry) Command



Purpose

The Change Configuration List Entry (CHGCFGLE) command changes one entry in the SNA pass-through configuration list. One entry consists of a group name, entry description, and upstream SNA pass-through device names. The user can add, remove, or change the SNA pass-through device names associated with each entry denoted by a group name.

Required Parameter

TYPE

Specifies the type of configuration list entry being changed.

***SNAPASTHR:** The SNA pass-through list is used. One SNA pass-through entry can be specified to be changed in the configuration list.

Optional Parameter

SNAPASTHRE

Specifies the SNA pass-through entry. This parameter is required if TYPE(*SNAPASTHR) is specified. One entry can be specified for this parameter.

The following values make up a SNA pass-through entry:

Element 1: SNA Upstream Group Name

group-name: Specify the name of the entry being changed. An entry is identified by a group name. The group name must already exist in the configuration list. The group name has upstream SNA pass-through device names associated with it.

Element 2: SNA Upstream Device Name

***SAME:** The value does not change. *SAME can only be specified once. No additions, changes, or removals of upstream device names are made.

SNA-pass-through-device-name: Specify the name of the device to use for the pass-through session.

Element 3: Text Description

***SAME:** The value does not change.

***BLANK:** Text is not specified.

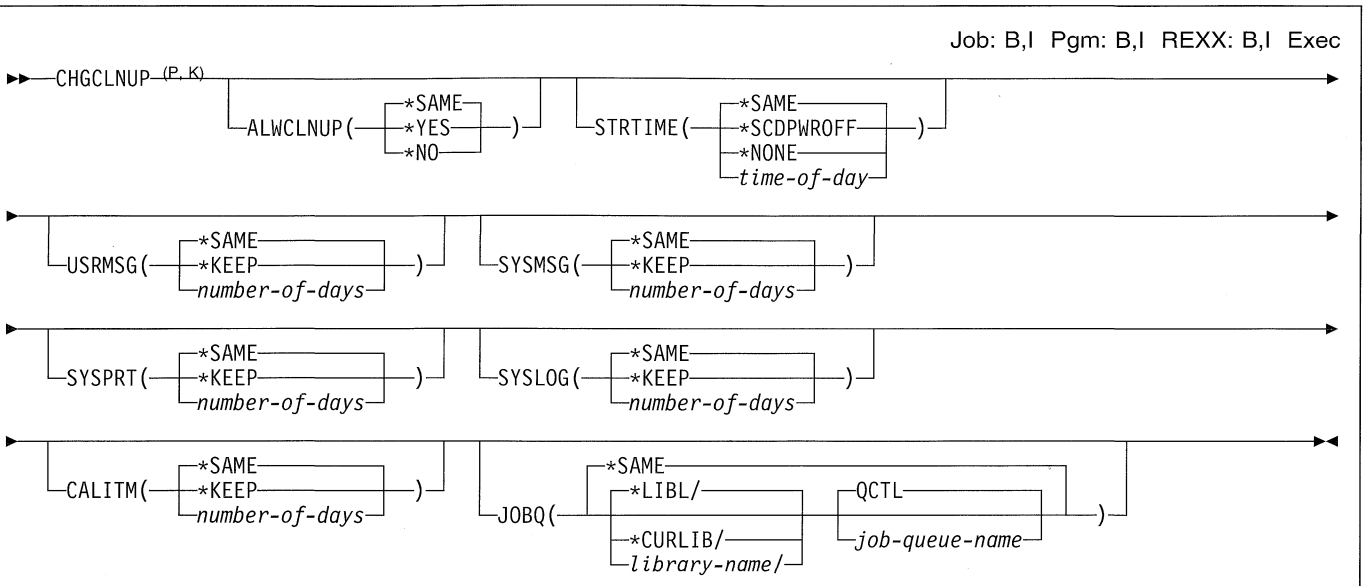
'entry-description': Specify a description of up to 50 characters for the SNA pass-through entry being changed.

Example

```
CHGCFGLE  TYPE(*SNAPASTHR)
          SNAPASTHRE(CICSGROUP
          (HOSTDEV1  HOSTDEV2)'Chicago Host Cics group')
```

This command changes the device names currently associated with the group to only those specified in the command.

CHGCLNUP (Change Cleanup) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Cleanup (CHGCLNUP) command allows the user to specify the cleanup options. The cleanup options control the following:

- If cleanup is allowed to run.
- When cleanup is run each day.
- What objects are cleaned up.

If cleanup is not active when this command is entered, the new values specified for this command are used the next time cleanup is started (with the Start Cleanup (STRCLNUP) command or as described in the *Operator's Guide*).

Restriction: The user must have *ALLOBJ and *JOBCTL authorities to use this command.

Optional Parameters

ALWCLNUP

Specifies whether cleanup can be run on the system. If ALWCLNUP(*YES) is specified, cleanup can be started with the STRCLNUP command or as specified in the *Operator's Guide*.

***SAME:** The value does not change.

***YES:** Cleanup can be run on this system.

***NO:** Cleanup cannot be run on this system.

STRTIME

Specifies the time when cleanup will run each day.

***SAME:** The value does not change.

***SCDPWROFF:** Cleanup starts at the time of the scheduled power off. The power off takes place when cleanup finishes, whether or not cleanup was completed successfully.

***NONE:** No cleanup start time is scheduled. The cleanup batch jobs are not submitted.

time-of-day: Specify the time when daily cleanup will run each day.

USRMSG

Specifies whether messages on user profile message queues are cleaned up. The cleanup function deletes messages on the user message queues that have remained on the system longer than the number of days specified.

***SAME:** The value does not change.

***KEEP:** Messages are not deleted.

number-of-days: Specify the number of days messages are kept before they are deleted. Valid values range from 1 through 366.

SYSMSG

Specifies that messages on the QSYSOPR message queue and on work station message queues are cleaned up. The cleanup function deletes messages on the QSYSOPR message queue and on work station message queues that have remained on the system longer than the number of days specified.

***SAME:** The value does not change.

***KEEP:** Messages are not deleted.

number-of-days: Specify the number of days messages are kept before they are deleted. Valid values range from 1 through 366.

SYSPRT

Specifies that job logs and other system output are cleaned up.

To prevent this output from being mixed with the user's output, the output queue of the printer file for job logs (QPJOBLOG) is changed to QUSRSYS/QEZJOBLOG to receive the job log. The output queues, QPPGMDMP, QPBASDMP, and QPSRVDMP are changed to QUSRSYS/QEZDEBUG. All entries in QEZJOBLOG and QEZDEBUG that are older than the number of days specified on this parameter are deleted.

If the cleanup operation is ended, the output queues named QEZJOBLOG and QEZDEBUG continue to be used for job logs, service dumps, and program dumps.

***SAME:** The value does not change.

***KEEP:** Job logs and other system output are not deleted.

number-of-days: Specify the number of days job logs and other system output are kept before they are deleted. Valid values range from 1 through 366.

SYSLOG

Specifies that system journal receivers, history log files, problem log files and problem log entries, alert database entries, and program temporary fixes are cleaned up.

Journal receivers:

Journal receivers that are used for one of the following system journals and are older than the number of days specified on this parameter are cleaned up:

QAOSDIAJRN	Journal for DIA files
QDSNX	Journal for DSNX logs
QSNADS	Journal for SNADS files
QSXJRN	Journal for problem databases
QPFRADJ	Journal for performance adjustment data
QACGJRN	Journal for job accounting data
QX400	Journal for X400 files

Note: The journal receiver for job accounting (QACGJRN) is cleaned up only if the journal is created by Operational Assistant*.

History log files:

History log files that meet both of the following conditions are deleted:

1. History log files that are older than the number of days specified on this parameter.
2. History log files named QSYS/QHST*.

Problem log files and problem log entries:

Any problem log entries older than the number of days specified on this parameter are deleted. The Delete Problem (DLTPRB) command is run to delete the problem log entries. When the DLTPRB command is run, the number of days specified for this parameter is used on the DAYS parameter of the DLTPRB command. The defaults are used for all other parameters on the DLTPRB command. Refer to the DLTPRB command description for more details on how this command runs.

Note: If the number of days specified for this parameter is less than the number of days specified for the system value QPRBHLDITV (Problem Log Hold Interval), the value for QPRBHLDITV is used instead for cleaning up problem logs.

In addition to the problem log entries being deleted, the following problem log files are reorganized:

Note: The following files are in library QUSRSYS.

QASXCALL	QASXFRU	QASXNOTE
QASXPROB	QASXPTF	QASXYMP
QASXEVT		

Alert database entries: Any alert database entries older than the number of days specified on this parameter are deleted. The Delete Alert (DLTALR) command is run to delete the alert database entries. When the DLTALR command is run, the number of days specified for this parameter is used on the DAYS parameter of the DLTALR command. The defaults are used for all other parameters on the DLTALR command. Refer to the DLTALR command description for more details on how this command runs.

In addition to the alert database entries being deleted, the file QUSRSYS/QAALERT is reorganized.

Program temporary fixes:

The following PTFs are deleted:

- Temporary objects named:
  - QPZA000000 through QPZA999999
  - QPZR000000 through QPZR999999
  - QPZI000000 through QPZI999999
  - QSCA000000 through QSCA999999
  - QSCR000000 through QSCR999999
- Exit programs shipped with PTFs
- Physical files in QUSRSYS
- QAPZPTF
- QAPZREQ
- QAPZSYM

If the library, QSMU, exists on the system, then only the PTFs for the current release are cleaned up. If the QSMU library does not exist on the system, then PTFs for this and all previous releases are cleaned up.

***SAME:** The value does not change.

***KEEP:** System journals and system logs are not deleted.

number-of-days: Specify the number of days system journals and system logs are kept before they are deleted. Valid values range from 1 through 366.

CALITM
Specifies whether OfficeVision/400* calendar items are kept or deleted by the cleanup operation after a specified number of days.

***SAME:** The value does not change.

***KEEP:** Calendar items are not deleted.

number-of-days: Specify the number of days calendar items are kept before they are deleted. Valid values range from 1 through 366.

JOBQ
Specifies the qualified name of the job queue on which this job is placed.

***SAME:** The job queue does not change.

The name of the job queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

QCTL: Batch cleanup jobs are submitted to the QCTL job queue.

job-queue-name: Specify the qualified name of the job queue on which the submitted job is placed.

Examples

Example 1: Keeping User Messages During Cleanup

```
CHGCLNUP  ALWCLNUP(*YES)  USRMSG(*KEEP)  STRTIME(0700)
```

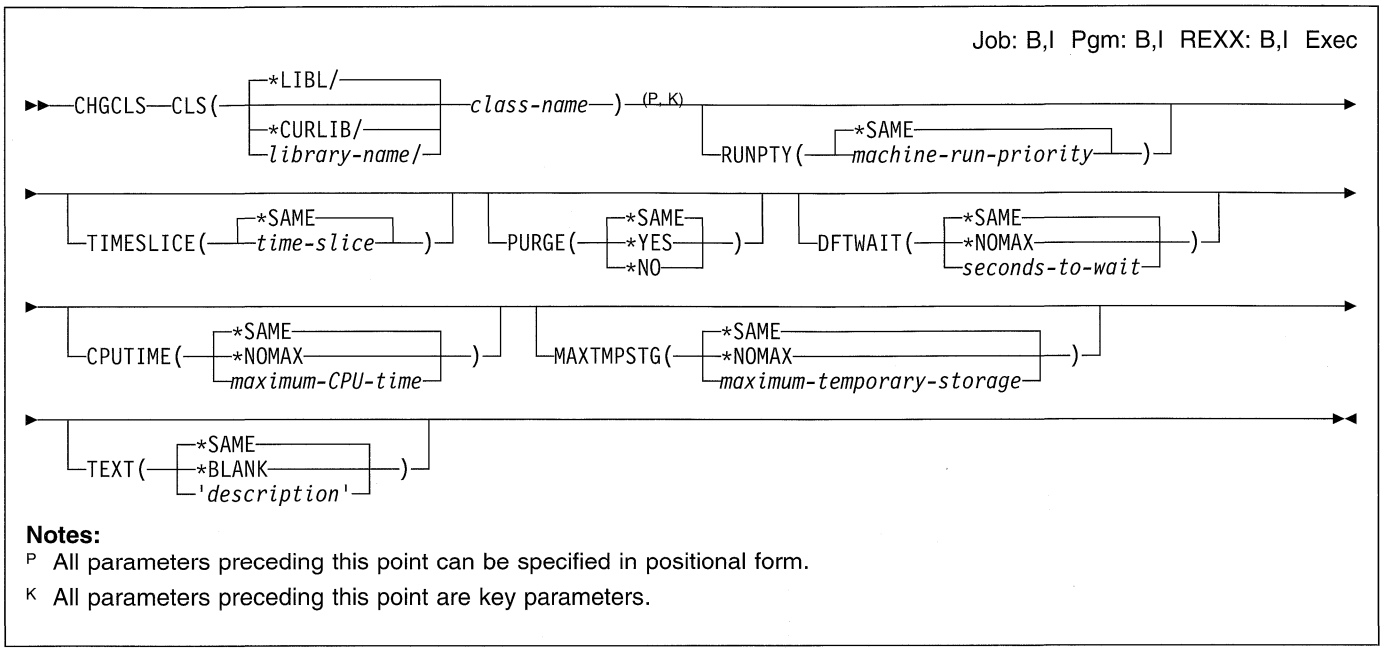
This command changes the cleanup options so that user messages are kept and not deleted when cleanup is performed. This command sets cleanup start time at 7:00 A.M.

Example 2: Keeping System Journals and System Logs During Cleanup

```
CHGCLNUP  ALWCLNUP(*YES)  SYSMSG(10)  SYSLOG(3)
```

This command changes the cleanup options so that system messages are kept for ten days, and system journals and system logs are kept for three days, before being deleted.

CHGCLS (Change Class) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Class (CHGCLS) command changes a class object that is specified in the attributes contained in a class. The class defines the processing parameters for jobs that use the class. The class used by a job is specified in the subsystem description routing entry used to start a job.

Required Parameter

CLS

Specifies the qualified name of the class.

Note: The following IBM-supplied objects are not valid on this parameter: QARBCLS, QLPINSTALL, and QMONCLS.

More information on this parameter is in "Appendix A. Expanded Parameter Descriptions" in the *CL Reference*.

The name of the class can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

class-name: Specify the name of the class.

Optional Parameters

RUNPTY

Specifies the running priority of jobs that use the class being changed. Machine running priority is a value, ranging from 1 (highest priority) through 99 (lowest priority), that represents the importance of the job when it competes with other jobs in the class for the machine resources. This value represents the relative importance, not the absolute importance, of the job. For example, a job with a running priority of 25 is not twice as important as one with a running priority of 50.

***SAME:** The value does not change.

machine-run-priority: Specify the running priority of the job that uses this specified class.

TIMESLICE

Specifies the maximum amount of processor time (in milliseconds) given to a job using this class before other jobs are given the opportunity to run. The time slice establishes the amount of time needed by the job to accomplish a meaningful amount of processing. At the end of the time slice, the job can be put in an inactive state so that other jobs can become active in the storage pool.

***SAME:** The value does not change.

time-slice: Specify the maximum number of milliseconds that each job that uses this class can have to run. Valid values range from 1 through 9999999 (that is, 9,999,999 milliseconds or 9999.999 seconds).

PURGE

Specifies whether the job is marked as eligible to be moved out of main storage and put into auxiliary storage

at the end of a time slice or when there is a long wait (such as waiting for a work station user's response).

***SAME:** The value does not change.

***YES:** The job is eligible to be moved out of main storage and put into auxiliary storage.

***NO:** The job is not eligible to be moved out of main storage and put into auxiliary storage. However, when some of main storage is needed for other jobs running in the same storage pool, pages belonging to this job are moved (eight at a time) to auxiliary storage to accommodate pages needed for other jobs. Then, when this job runs again, its pages are returned to main storage as they are needed.

DFTWAIT

Specifies the default maximum wait time (in seconds) that processing of a job is delayed until a system instruction that performs a wait finishes running. This default wait time is used when a wait time is not otherwise specified for a given situation. Normally, this would be the amount of time the system user is willing to wait for the system before the request is canceled.

If the wait time for an instruction is exceeded, an error message is either displayed or it can be automatically handled by a Monitor Message (MONMSG) command.

***SAME:** The value does not change.

***NOMAX:** There is no limit on the time the system waits for the completion of the operation.

seconds-to-wait: Specify a value, ranging from 0 through 9999999 seconds, that specifies the maximum time the system waits for the system instruction to finish running.

Note: Although a 0 default wait time is allowed, it is not recommended. Some system instructions require the use of system resources that may be in use and with a 0 default time, will cause the instruction to fail. When a system instruction fails (exceeds the default wait time) unexpected results may occur for the job. Most system resources will only be in use for a short time, so having a small default wait time will not noticeably degrade the performance of the job.

CPUTIME

Specifies the maximum processing unit time (in milliseconds) the routing step of a job using this class can have to completely run the job. If not finished before the maximum time is used, running of the job is ended.

***SAME:** The value does not change.

***NOMAX:** There is no limit on the time the system waits for the completion of the operation.

maximum-CPU-time: Specify the maximum number of milliseconds that the routing step of the job has in which to finish running. Valid values range from 1 through 9999999 seconds.

MAXTMPSTG

Specifies the maximum amount of temporary (auxiliary) storage (in kilobytes) that a job in this class can use for processing. This temporary storage is used for storage required by the program itself and by implicitly created internal system objects used to support the job. (It is not storage in the QTEMP library.) If the maximum temporary storage is exceeded by a job, the job is ended. This parameter does not apply to the use of permanent storage, which is controlled through the user profile.

***SAME:** The value does not change.

***NOMAX:** There is no maximum amount of temporary storage for the job that uses this class.

maximum-temporary-storage: Specify the number of kilobytes (ranging from 0 through 9999999) that specifies the maximum amount of temporary storage that a job in this class can have.

TEXT

Specifies text that briefly describes the class. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

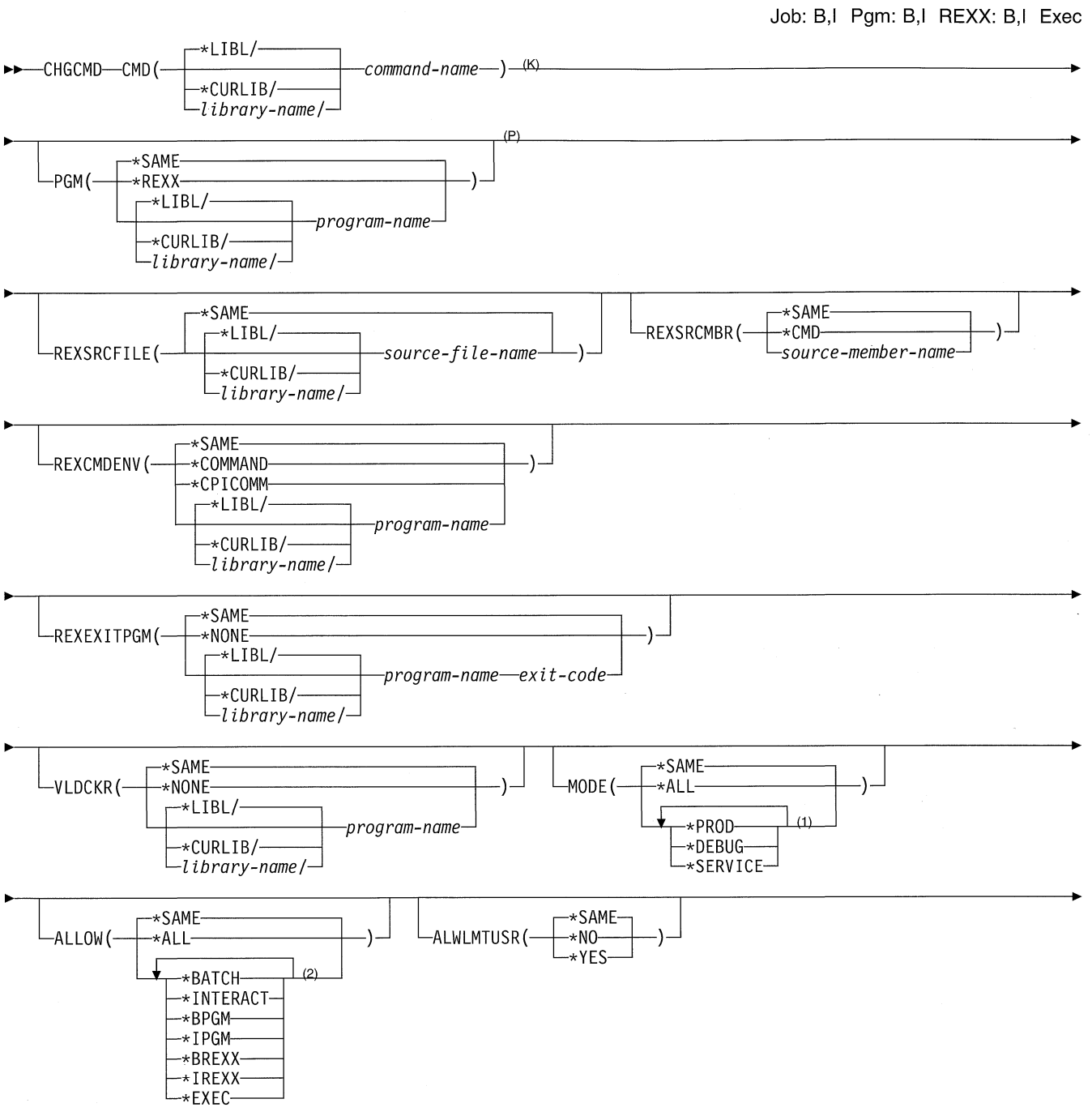
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGCLS  CLS(CLASS1)  RUNPTY(60)  TIMESLICE(900)
```

This command changes a class called CLASS1 in the library on the job's library list. The run priority for the class is changed to 60 and a time slice of 900 milliseconds

CHGCMD (Change Command) Command



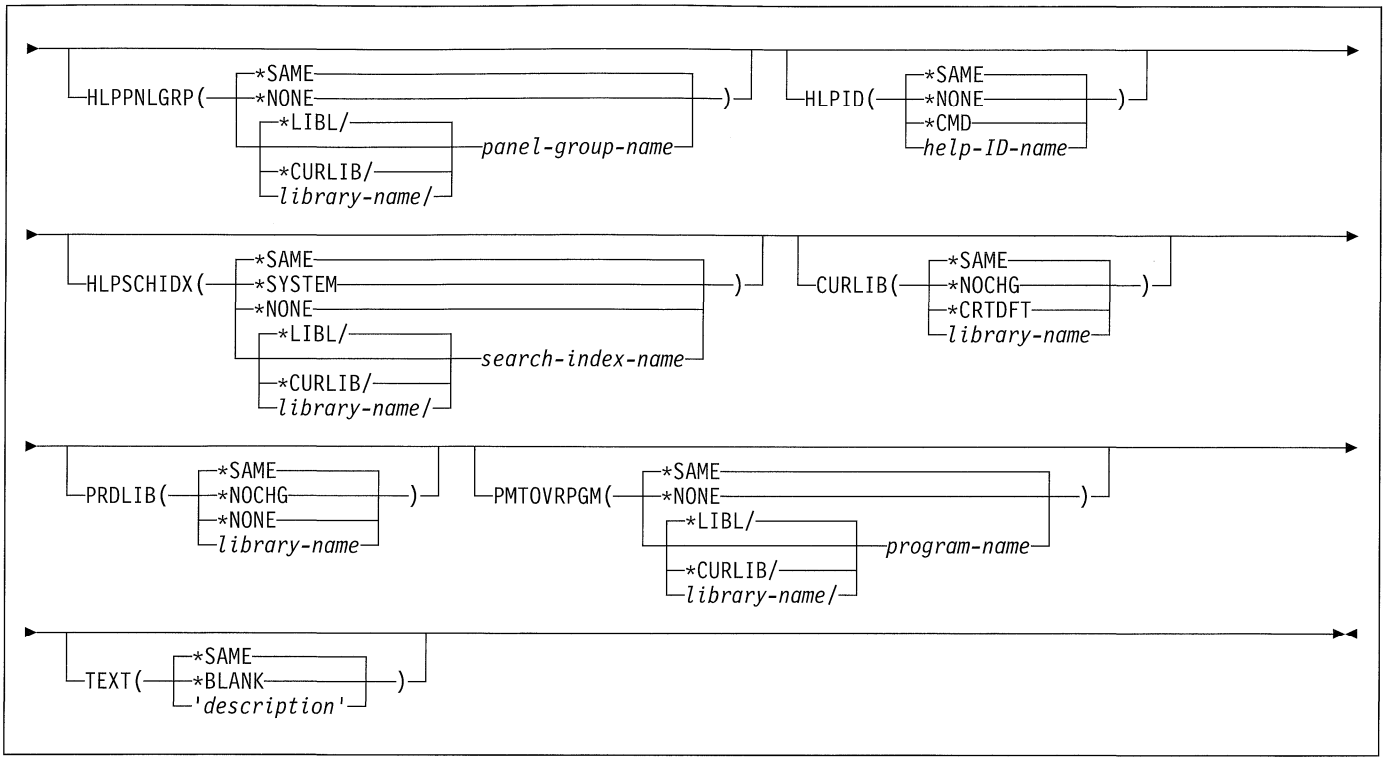
Notes:

K All parameters preceding this point are key parameters.

P All parameters preceding this point can be specified in positional form.

1 A maximum of 3 repetitions

2 A maximum of 7 repetitions



Purpose

The Change Command (CHGCMD) command changes some of the attributes of a command definition. It can specify a different command processing program (CPP) to process the command; it can also change the type of operating environment (production, debug, or service) in which the command can be processed and the text description of the command. CL programs that use the command being changed by the CHGCMD command do *not* have to be re-created. The CHGCMD command does not change the parameter descriptions or validity checking information in the command definition object.

Restrictions:

1. The user must have object management authority for the command that is being changed.
2. The CHGCMD command can be used to change only the attributes of a created CL command. That is, those attributes that were specified on the Create Command (CRTCMD) command. The CHGCMD command cannot be used to change attributes of *statements*, such as command definition statements.
3. The CHGCMD should not be used to change the command processing program (PGM parameter), the validity checking program (VLDCKR parameter), or the prompt override program (PMTOVRPGM parameter) of an IBM-supplied command.

Required Parameter

CMD

Specifies the qualified name of the command being changed. The command can be either a user-defined command or IBM-supplied command.

The name of the command can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

command-name: Specify the name of the command being changed.

Optional Parameters

PGM

Specifies the qualified name of the command processing program (CPP) that processes the command.

***SAME:** The value does not change.

***REXX:** The CPP for this command is the REXX procedure identified on the REXSRCMBR and REXSRCFILE parameters.

The name of the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the CPP processing the command.

REXSRCFILE

Specifies the qualified name of the REXX source file containing the REXX procedure that is the command processing program (CPP).

***SAME:** The REXX source file does not change.

The name of the REXX source file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

source-file-name: Specify the name of the source file that contains the source member that is used.

REXSRCMBR

Specifies the source member containing the REXX procedure that functions as the command processing program (CPP).

***SAME:** The value does not change.

***CMD:** The source member name is the same as the command name (the name specified on the CMD parameter).

source-member-name: Specify the name of the source member that contains the REXX procedure.

REXCMDENV

Specifies the command environment that is active when the REXX CPP starts to run. If the AS/400 control language (CL) environment is not used, a program can be called to set the environment. The REXX interpreter calls this program to process commands encountered in the procedure. This environment can be changed with the REXX ADDRESS instruction.

***SAME:** The value does not change.

***COMMAND:** The AS/400 control language (CL) command environment is used.

***CPICOMM:** The Common Programming Interface (CPI) for Communications command environment is used. CPICOMM is the command environment used for CL commands that are imbedded within a REXX procedure.

The name of the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the program to process commands encountered by the REXX program.

REXEXITPGM

Specifies the exit program to be used when the REXX interpreter is started. This parameter can be specified as a single value (*SAME or *NONE) or as a list of two values (elements).

***SAME:** The value does not change. ***NONE:** There are no REXX interpreter exit programs for this call of the CPP.

The name of the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

Element 1: Name of the Exit Program

program-name: Specify the name of the exit program.

Element 2: Value of the Exit Code

exit-code: Specify one of the following values for the exit code:

Exit-code	Description
2	The associated program is called whenever an external function or subroutine has been called by the REXX program. The exit program is then responsible for locating and calling the requested routine.
3	The associated program is called whenever the interpreter is going to call a command. The exit program is responsible for locating and calling the command given the command string and the current environment name.
4	The associated program is called whenever a REXX instruction or function attempts an operation on the REXX external data queue.
5	The associated program is called when session input or output operations are attempted.

7 The associated program is called after running each clause of the REXX procedure to determine whether it should be halted.

8 The associated program is called after running each clause of the REXX program to check whether tracing should be turned on or off.

9 The associated program is called before interpretation of the first instruction of a REXX procedure (including REXX procedures called as external functions and subroutines).

10 The associated program is called after interpretation of the last instruction of a REXX procedure (including REXX procedures called as external functions and subroutines).

VLDCKR

Specifies the qualified name of a program that does additional validity checking on the parameters in the command being processed. The validity checker is called to do additional user-defined validity checking beyond that specified by the command definition statements in the source file, and beyond the syntax checking that is done on the command when it is compiled.

***SAME:** The value does not change.

***NONE:** There is no separate validity checking program for this command. All validity checking is done by the command analyzer and the command processing program. Whenever the command is processed or checked for validity, provided variables and expressions are not used.

The name of the validity checking program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the validity checker program that checks the validity of the command.

MODE

Specifies the types of operating environments in which the Change Command (CHGCMD) command can be used. One or more of the modes can be specified.

***SAME:** The value does not change.

***ALL:** The command is valid in all types of operation: the production, debug, and service modes.

***PROD:** The command is valid for the production mode.

***DEBUG:** The command is valid for the debug mode.

***SERVICE:** The command is valid for the service mode.

ALLOW

Specifies where the command can be processed. One or more of the following options can be specified.

***SAME:** The value does not change.

***ALL:** The command is valid in a batch input stream, in a CL program, in a REXX procedure, or when processed interactively. It can also be passed to the system program QCMDEXC (or QCAEXEC) for processing.

***BATCH:** The command can be processed in a batch input stream that is external to a compiled CL program.

***INTERACT:** The command is valid when processed interactively, external to a compiled CL program.

***BPGM:** The command can be processed in a compiled CL program that is called from batch entry.

***IPGM:** The command can be processed in a compiled CL program that is called from an interactive entry.

***BREXX:** The command can be used in a REXX procedure that is run in batch job.

***IREXX:** The command can be used in a REXX procedure that is run in an interactive job.

***EXEC:** The command can be used as a parameter on the CALL command and be passed as a character string to the system program QCMDEXC (or QCAEXEC) for processing. If *EXEC is specified, either *BATCH or *INTERACT must also be specified.

ALWLMTUSR

Specifies whether the command can be entered from the command line on a menu by a user whose profile is set for limited capabilities. (LMTCPB keyword on the Create User Profile (CRTUSRPRF) and Change User Profile (CHGUSRPRF) commands.)

***SAME:** The value does not change.

***NO:** This command cannot be entered from the command line on a menu by a user whose profile is set for limited capabilities.

***YES:** This command can be entered from the command line on a menu by a user whose profile is set for limited capabilities.

HLPPNLGRP

Specifies the qualified name of the help panel group for this command.

***SAME:** The value does not change.

***NONE:** No help panel group is specified.

Note: If *NONE is specified on this parameter and a value is specified on the HLPID parameter, an error message is issued when the command is run.

The name of the panel group can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

panel-group-name: Specify the name of the help panel group for this command.

HLPID

Specifies the name of the general help module for the help identifiers for this command.

***SAME:** The value does not change.

***NONE:** No help identifier is specified.

Note: If *NONE is specified on this parameter and a value is specified on the HLPPNLGRP parameter, an error message is issued when the command is run.

***CMD:** The name of the command is used as the first half of all the names for help IDs.

help-ID-name: Specify the name of the general help module for the help identifiers for this command.

HLPSCHIDX

Specifies the qualified name of the help search index to use when the search index function key is pressed from the help screen.

***SAME:** The value does not change.

***SYSTEM:** The system search index, QHSS1, is used.

***NONE:** No help search index is associated with this command.

The name of the search index can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

search-index-name: Specify the name of the search index to be used when the search index function key is pressed.

CURLIB

Specifies the name of the library being used as the current library during the processing of this command.

Note: This library is also the current library when the validity checking program (if any) is processed for the command.

***SAME:** The value does not change.

***NOCHG:** The current library is not changed for the processing of this command. If the current library is changed during processing of the command, the change remains in effect after command processing is complete.

***CRTDFT:** There is no current library active during processing of the command. The current library that was active before command processing is restored when processing is completed.

If *CURLIB is specified as the to-value for any single values or special values for this command, or for any command processed while there is no current library active, the QGPL library is used as the current library.

library-name: Specify the name of the library used as the current library. The library need not exist when the command is created, but must exist when the command is processed. When command processing is completed, the current library is restored to its previous value. If the current library is changed during command processing by the Change Library List (CHGLIBL) command or Change Current Library (CHGCURLIB) command, the change is effective only until the command is processed. QTEMP cannot be specified for the current library.

PRDLIB

Specifies the name of the product library that is in effect during the processing of the command.

***SAME:** The value does not change.

***NOCHG:** The product library does not change when processing of the command starts. If the product library is changed during the processing of the command, the change remains in effect after command processing is complete.

***NONE:** There is no product library in the job's library list. The product library is restored to its previous value when command processing is completed.

library-name: Specify the name of the library used as the current library. The library need not exist when the command is created, but must exist when the command is processed. When command processing is completed, the current library is restored to its previous value. If the current library is changed during command processing by the Change Library List (CHGLIBL) command or Change Current Library (CHGCURLIB) command, the change is effective only until the command is processed. QTEMP cannot be specified for the current library.

PMTOVRPGM

Specifies the qualified name of the prompt override program (POP) that replaces (on the prompt display) the default values with the current actual values for the parameter. If a POP is specified, the key parameters (specified as KEYPARM(*YES) on the PARM statement in the command definition source) are the only parameters shown on the initial prompt display. When values are input for the key parameters, the remaining parameters are shown on the screen with the actual values instead of the default values.

***SAME:** The value does not change.

***NONE:** No prompt override program is specified.

Note: If *NONE is specified when key parameters exist in the command definition source (KEYPARM(*YES) specified on the PARM statement), a warning message is issued when the command is created, and KEYPARM(*NO) is assumed for all parameters.

The name of the prompt override program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the prompt override program.

TEXT

Specifies text that briefly describes the command. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** No text is specified.

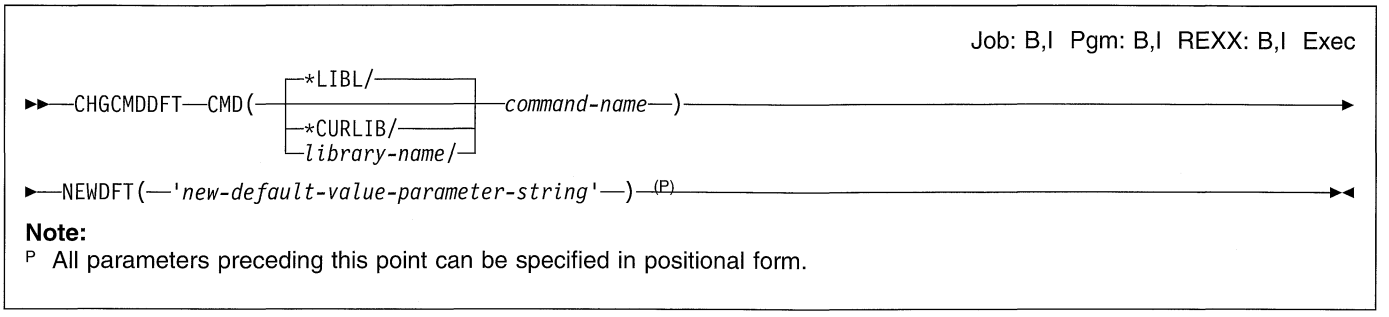
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGCMD  CMD(PAYROLL)  VLDCKR(LIB01/PAYVLDPGM)
```

The validity checking program for the PAYROLL command is the program named PAYVLDPGM located in library LIB01. All other attributes of the PAYROLL command remain the same.

CHGCMDDFT (Change Command Default) Command



Note:
P All parameters preceding this point can be specified in positional form.

Purpose

The Change Command Default (CHGCMDDFT) command changes the default value of a command parameter. The parameter must have an existing default to change to a new default value. The changed command can be either an IBM-supplied command or a user-defined command.

To find out which parameters of a command have default values, enter the command name and press the F4 key (Prompt) to prompt for the command. Default values for parameters are shown on the first command prompt screen. However, not all defaults are shown. A parameter that is a list of elements or a qualified name may have a single value (SNGVAL) as the default of the entire list or qualified name. For example, Display Job (DSPJOB) is a command in which JOB is a list of elements but has a SNGVAL as the default for the entire list. See 'Additional Considerations' for more information on SNGVAL as a default value.

The new parameter default must be valid for the parameter being changed. To find out which values are valid, move the cursor to the default value shown and press the F4 key. The Parameter Prompt with Permissible Values display shows a list of valid values and/or the value of the parameter type. When only a list of valid values is shown, only those values can be used as new default values. Otherwise, any valid value that conforms to the requirements of the parameter can be used as the new default.

Processing a command that had earlier default value changes is the same as specifying the new default value for the parameter on an unchanged copy of the same command. The new default value may conflict with other parameter combinations even when the parameter is not explicitly specified.

When the user runs system commands with default value changes from system displays, the new default values cannot be used. This occurs because the system cannot use the command device to process a function. Instead, the system can use an IBM internal device to a system program to run the requested function. Therefore, use the F4 key (Prompt) key to prompt for the system command when processing the command from a system display. In most cases, the specified default value is then used.

To change an IBM-supplied command, the user should create a copy of the command in a user library and change the defaults of the copied command, instead of the IBM-supplied command itself. The user library should be listed before the library QSYS on the library list. Save the source of the CHGCMDDFT command so that the changes can be identified later. This procedure allows the user to use the original IBM-supplied command defaults when necessary and ensures that changes made to the copied commands are not overlaid when a new release on the system is installed.

A compatibility problem can occur when a new release of the system is installed. If IBM-supplied commands have been copied into a user library and the newly installed IBM-supplied version of the command has had new parameters added to it, the copied version of the command does not run correctly. This is a result of parameter mismatches between the copied command and the newly installed IBM-supplied command processing program. Also, if a copied command is used in the starting program of a user profile, this program does not run correctly, again because of parameter mismatches between the old command and the new command processing program. As a result, the user cannot sign onto the system. To prevent this, the user should ensure that at least one user profile exists on the system which has authority to make the required changes to the starting programs and has QCMD as the starting program. An alternative is to code the starting programs to detect all errors and to recover by showing the Command Entry display (program QCMD).

The *CL Programmer's Guide* contains a list of recommendations and examples for using the Change Command Default (CHGCMDDFT) command.

Restriction: The user must have object management and *USE authority for the command being changed.

Required Parameters

CMD
Specifies the qualified name of the command being changed. The command can be a user-defined command or an IBM-supplied command.

The name of the command can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

command-name: Specify the name of the command being changed.

NEWDFT

Specifies the parameter string, enclosed in apostrophes, that contains the new default values being assigned to a parameter, element, or qualifier if a value is not specified by the user when the command is processed.

New default values may be specified only for those parameters, elements, or qualifiers that already have a default value.

*N must be used as a place holder in a qualified name or in a list of values to take the place of elements or qualifiers that do not have existing default values. If the elements or qualifiers have default values but are not to be changed to a new default value, then *N or the existing default value can be used as the place holder in the list of values or the qualified name.

```
CHGCMDDFT CMD (USRQSYS/CRTCLPGM)
  NEWDFT('PGM(USERLIB1/*N)')
            or
CHGCMDDFT CMD (USRQSYS/DSPFD)
  NEWDFT('OUTMBR(*N *ADD)')
```

The new default value for each changed parameter, element, or qualifier must satisfy one of the following:

- It must match the requirements specified by TYPE, LEN, REL, RANGE, RSTD, and FULL attributes of the changed parameter, element, or qualifier.
- It must be a valid SPCVAL, SNGVAL, or VALUES value of the changed parameter, element, or qualifier.

A description of the TYPE, LEN, REL, RANGE, RSTD, FULL, SPCVAL, SNGVAL, or VALUES parameters is in Chapter 5, "Command Definition Statements," (PARM, ELEM, or QUAL).

Note: Checking between parameters and Validity Checking Program processing is done when the command is run. Neither are performed during the changing of the default value. However, if dependencies between parameters exist for the keyword being changed, an informational message is sent indicating that checking was not done for the keyword. The command should be run immediately after making the change in default value to determine whether any additional rules were violated. To return to the IBM-supplied default, specify the desired default on the CHGCMDDFT command.

Examples

The following examples show changes in the defaults of command parameters when the CHGCMDDFT command is processed.

Example 1: Changing Default Value of AUT Parameter

```
CHGCMDDFT  CMD(CRTPF) NEWDFT('AUT(*EXCLUDE)')
```

This command changes the AUT default from *CHANGE to *EXCLUDE on the Create Physical File (CRTPF) command.

Example 2: Changing CRTCLPGM Command Defaults

```
CHGCMDDFT   CMD(USRQSYS/CRTCLPGM)
  NEWDFT('PGM(LIB001/*N)  SRCFILE(LIB001/FILE001)
  LOG(*YES)')
```

This command changes the defaults of the Create CL Program (CRTCLPGM) command when the NEWDFT parameter is specified in the form above. In the example, the default value of the LOG parameter is *YES. No default can be specified for the second qualifier (program-name) of the PGM parameter because no default value exists. Therefore, *N is used as the place holder for the second qualifier. The first qualifier (library-name) of the PGM parameter has a default value of LIB001. For the SRCFILE parameter, the first qualifier (library-name) has a default value of LIB001. The second qualifier (source-file-name) has a default value of FILE001.

Example 3: Changing Default Value of MAXMBRS Parameter

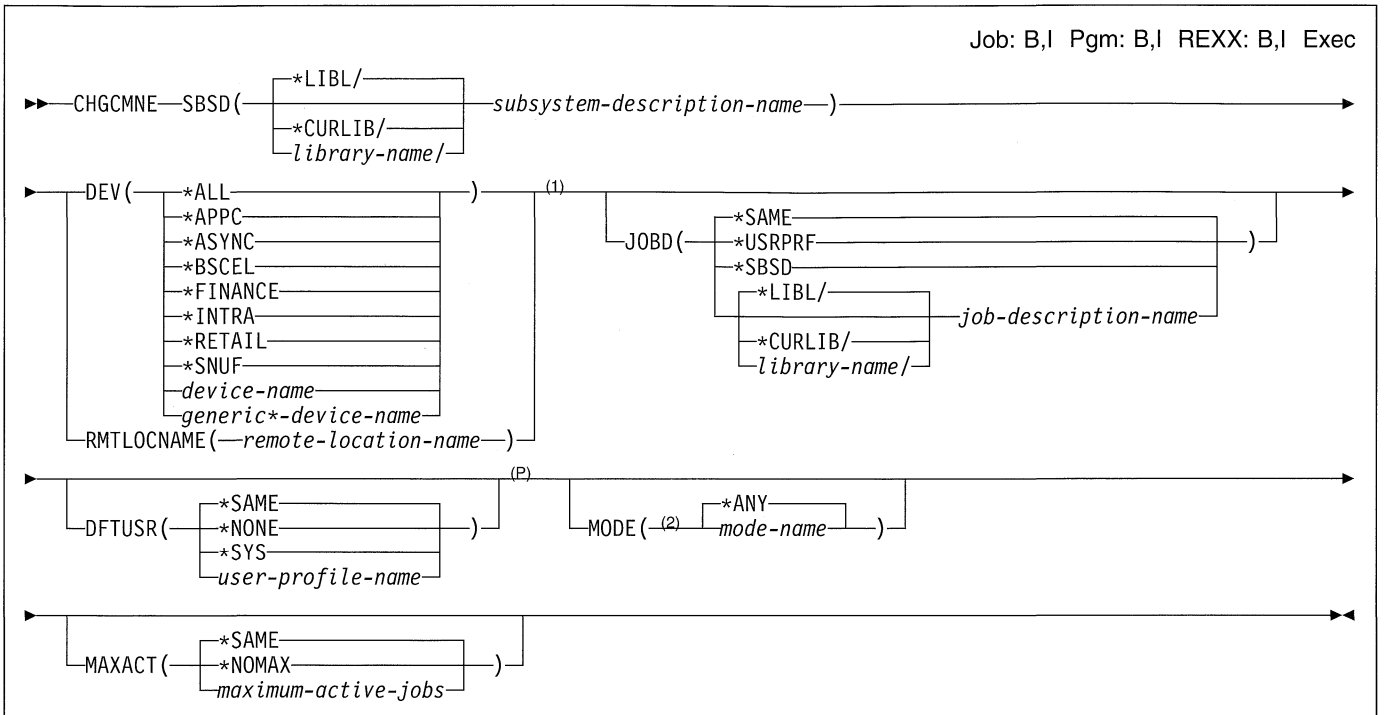
```
CRTPF  FILE(FILE1)  RCDLEN(96) MAXMBRS(1)
  .
  .
  .
CHGCMDDFT CMD(CRTPF) NEWDFT('MAXMBRS(*NOMAX)')
```

This command changes the default value of the MAXMBRS keyword on the CRTPF command to *NOMAX.

Additional Considerations

A keyword that is defined as a list of elements or as a qualified name, and has a SNGVAL as the default, can have a required element or qualifier as the first element in the list of elements or qualified name. The default SNGVAL may be changed to another valid SNGVAL if one exists, but the required element or qualifier cannot have a default value. A SNGVAL cannot be used as a new default value if the existing default value is *not* a SNGVAL. To find out if list elements or qualifiers of a qualified name have default values, change the SNGVAL to a valid value for the keyword, blank out remaining list elements or qualifiers, and press the Enter key. If any remaining list elements or qualifiers have default values, the default values now appear in the input field for the list element or qualifier.

CHGCMNE (Change Communications Entry) Command



Notes:

1 A user must specify the DEV parameter or the RMTLOCNAME parameter, but not both.

P All parameters preceding this point can be specified in positional form.

2 MODE(*ANY) must be specified if DEV(*ASYNC), DEV(*BSCEL), DEV(*FINANCE), DEV(*INTRA), DEV(*RETAIL), or DEV(*SNUF) is specified.

Purpose

The Change Communications Entry (CHGCMNE) command is used to change the attributes of a communications entry in an existing subsystem description. The subsystem must not be active when this command is entered.

Restriction: To use this command, the user must have object operational and object management authorities for the subsystem description and object operational authority for the job description.

Required Parameters

SBSD

Specifies the qualified name of the subsystem description that contains the communications entry to be changed.

The name of the subsystem description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

subsystem-description-name: Specify the name of the subsystem description that contains the communications entry being changed.

DEV

Specifies the name of the device description or the type of the device being used with this communications entry.

Note: A user must specify either this parameter or the RMTLOCNAME parameter, but not both.

***ALL:** All communications devices can be used with this communications entry.

***APPC:** All advanced program-to-program communications devices can be used with this communications entry. The devices created with the Create Device Description for APPC (CRTDEVAPPC) command are used.

***ASYNC:** All asynchronous communications devices can be used with this communications entry. The devices created with the CRTDEVASC command are used. This value is valid only if MODE(*ANY) is specified.

***BSCEL:** All bisynchronous equivalency link communications devices can be used with this communications entry. The devices created with the CRTDEVBSC command are used. This value is valid only if MODE(*ANY) is specified.

***FINANCE:** All FINANCE communications devices can be used with this communications entry. The devices created with the CRTDEVFNC command are used. This value is valid only if MODE(*ANY) is specified.

***INTRA:** All INTRA communications devices can be used with this communications entry. The devices created with the CRTDEVINTR command are used. This value is valid only if MODE(*ANY) is specified.

***RETAIL:** All RETAIL communications devices can be used with this communications entry. The devices created with the CRTDEVRTL command are used. This value is valid only if MODE(*ANY) is specified.

***SNUF:** All SNA upline facility communications devices can be used with this communications entry. The devices created with the CRTDEVSNUF command are used. This value is valid only if MODE(*ANY) is specified.

device-name: Specify the device description name or the type of device to use with this communications device entry. The name specified on the CRTDEVxxx command associated with this device description name is used.

generic-device-name:* Specify the generic name of the device. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be changed only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

RMTLOCNAME

Specifies the name of the remote location that is used with this object.

Note: The remote location name specified on the associated CRTDEVXXX command can be used here. The validity of the remote location name is not checked.

The user must specify either this parameter or the DEV parameter, but not both.

Optional Parameters

JOBD

Specifies the name of the job description used. If the job description does not exist when the entry is changed, a library qualifier must be specified because the qualified job description name is retained in the subsystem description. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***USRPRF:** The job description name specified in the user profile of the user that made the program start request is used by jobs that are processed through this communications entry.

***SBSD:** The job description having the same qualified name as the subsystem description, specified by the SBSD parameter, is used for jobs processed through this communications entry.

The name of the job description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

job-description-name: Specify the name of the job description that is used for the jobs processed through this communications entry.

DFTUSR

Specifies the default user profile used for a program start request that contains no password (all blanks or a zero length password) and user profile name in the request. This user profile is not used for program start requests that have either a password or user profile specified (either valid or not valid).

***SAME:** The value does not change.

***NONE:** No user profile is specified as the default.

***SYS:** User program start requests are treated the same as DFTUSR(*NONE). For system-generated program start requests, the correct user profile is used.

user-profile-name: Specify the name of the user profile that is used for all program start requests that enter the system through this communications entry and that contain no password or user profile information.

Note: The names QSECOFR, QSPL, QDOC, QDBSHR, QRJE, and QSYS are not valid entries for this parameter.

MODE

Specifies the mode name of the communications device or remote location whose communications entry is being changed.

***ANY:** Any available modes defined to the communications device or remote location are allocated to the subsystem. If the communications device does not have defined modes associated with it, the communications device itself is allocated to the subsystem.

mode-name: Specify a mode name of the communications device or remote location name being changed.

MAXACT

Specifies the maximum number of jobs (received program start requests) that can be active at the same time through this communications entry. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***NOMAX:** There is no maximum number of jobs that can be active at the same time.

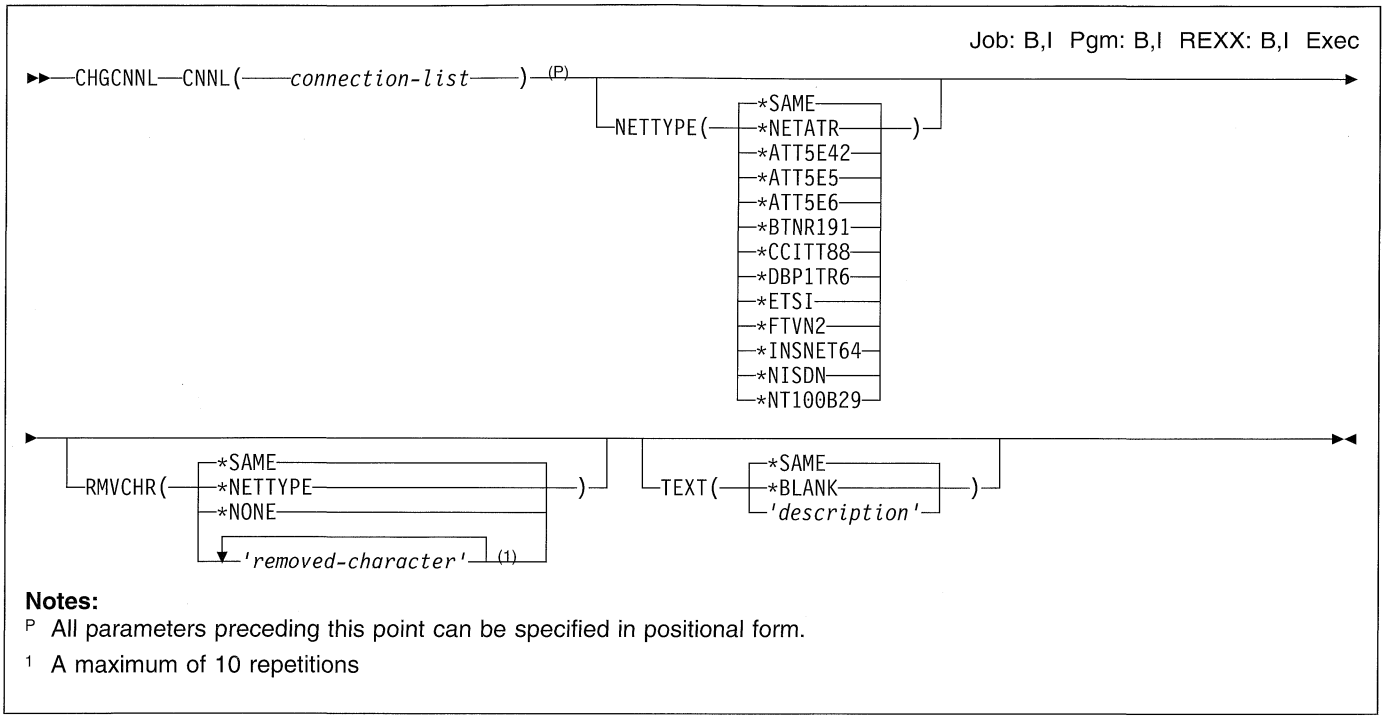
maximum-active-jobs: Specify the maximum number of jobs that can be active at the same time through this communications entry.

Example

```
CHGCMNE  SBSD(QGPL/BAKER)  DEV(A12)  MAXACT(*NOMAX)
```

This command changes the communications entry (in the subsystem description QGPL/BAKER) for the device A12. The maximum activity level is changed to *NOMAX which means that the communications entry puts no restrictions on the number of program start requests that may be active at the same time. However, the MAXJOBS value in the subsystem description BAKER limits the total number of jobs that can be active in the subsystem. This includes those created by program start requests. There is also a limit that the user can specify on the number of active jobs that can be routed through any particular routing entry (MAXACT). The limit specified in the routing entry may control the number of jobs using a particular pool or the recursion level of a particular program. In all cases, none of these limits can be exceeded as a result of processing a program start request.

CHGCNNL (Change Connection List) Command



Notes:

P All parameters preceding this point can be specified in positional form.

1 A maximum of 10 repetitions

Purpose

The Change Connection List (CHGCNNL) command changes a connection list. A connection list consists of up to 256 connection list entries. Connection lists are used to manage incoming and outgoing calls. Users specify a network type and the removal of characters for all of the entries of a list. To add entries to a connection list, use the Add Connection List Entry (ADDCNNLE) Command.

Required Parameter

CNNL

Specifies the name of the connection list.

Optional Parameters

NETTYPE

Specifies the type of Integrated Services Digital Network (ISDN) used by the connection list. This parameter is used to determine the defaults for other parameters which are network-dependent. Those parameters include: Remote Number Type, Remote Numbering Plan, Remote Subaddress Type, Local Number Type, Local Numbering Plan, Local Subaddress Type, Transit Network Plan, and Transit Network Type.

Note: The defaults that are set for those parameters can be viewed by using the Add Connection Lists Entry (ADDCNNLE) command, or by checking the *ISDN Guide*.

***SAME:** The value does not change.

***NETATR:** The network type specified by the default network type parameter (DFTNETTYPE) in the network attributes is used. The values of the parameters in the network attributes for the system can be displayed by using the Display Network Attributes (DSPNETA) command.

***ATT5E42:** Use this value when the user is attaching to an ISDN in the United States or Canada that uses AT&T 5ESS** version 5E4.2 switching equipment.

***ATT5E5:** Use this value when the user is attaching to an ISDN in the United States or Canada that uses AT&T 5ESS version 5E5 switching equipment.

***ATT5E6:** Use this value when the user is attaching to an ISDN in the United States or Canada that uses AT&T 5ESS version 5E6 switching equipment.

***BTNR191:** Use this value when the user is attaching to an ISDN in the United Kingdom controlled by British Telecomm.

***CCITT88:** The ISDN default values recommended by the International Telegraph and Telephone Consultative Committee (CCITT) in 1988 are used.

***DBP1TR6:** Use this value when the user is attaching to an ISDN controlled by Germany's PTT (Deutsche Bundespost 1TR6).

***ETSI:** The European Telecommunications Standards Institute (ETSI, also known as EuroISDN) standard is used.

***FTVN2:** Use this value when the user is attaching to version 2 of the ISDN controlled by France's post tele-

phone and telegraph administration (PTT) (France Telecom Numeris VN2).

***INSNET64:** Use this value when the user is attaching to the INSNET64 ISDN controlled by Japan's Nippon Telephone and Telegraph Public Corporation (NTT).

***NISDN:** The National ISDN-1 or National ISDN-2 standard for North America is used.

***NT100B29:** Use this value when the user is attaching to an ISDN in the United States or Canada that uses Northern Telecom DMS100 Version BCS-29 switching equipment.

RMVCHR

Specifies up to 10 characters that should be removed from remote and local numbers before the numbers are used by the system. Extra characters are removed from numbers before sending or comparing those numbers. The ability to shorten numbers prior to their use by the system means the user can insert extra characters in numbers to help the user organize and read them.

***SAME:** The value does not change.

***NETTYPE:** A default value is used based on the network type specified on the NETTYPE parameter when the connection list was created using the Create Connection List (CRTCNNL) command.

***NONE:** No characters are removed.

'character': Specify up to ten characters to be removed.

TEXT

Specifies text that briefly describes the connection list. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

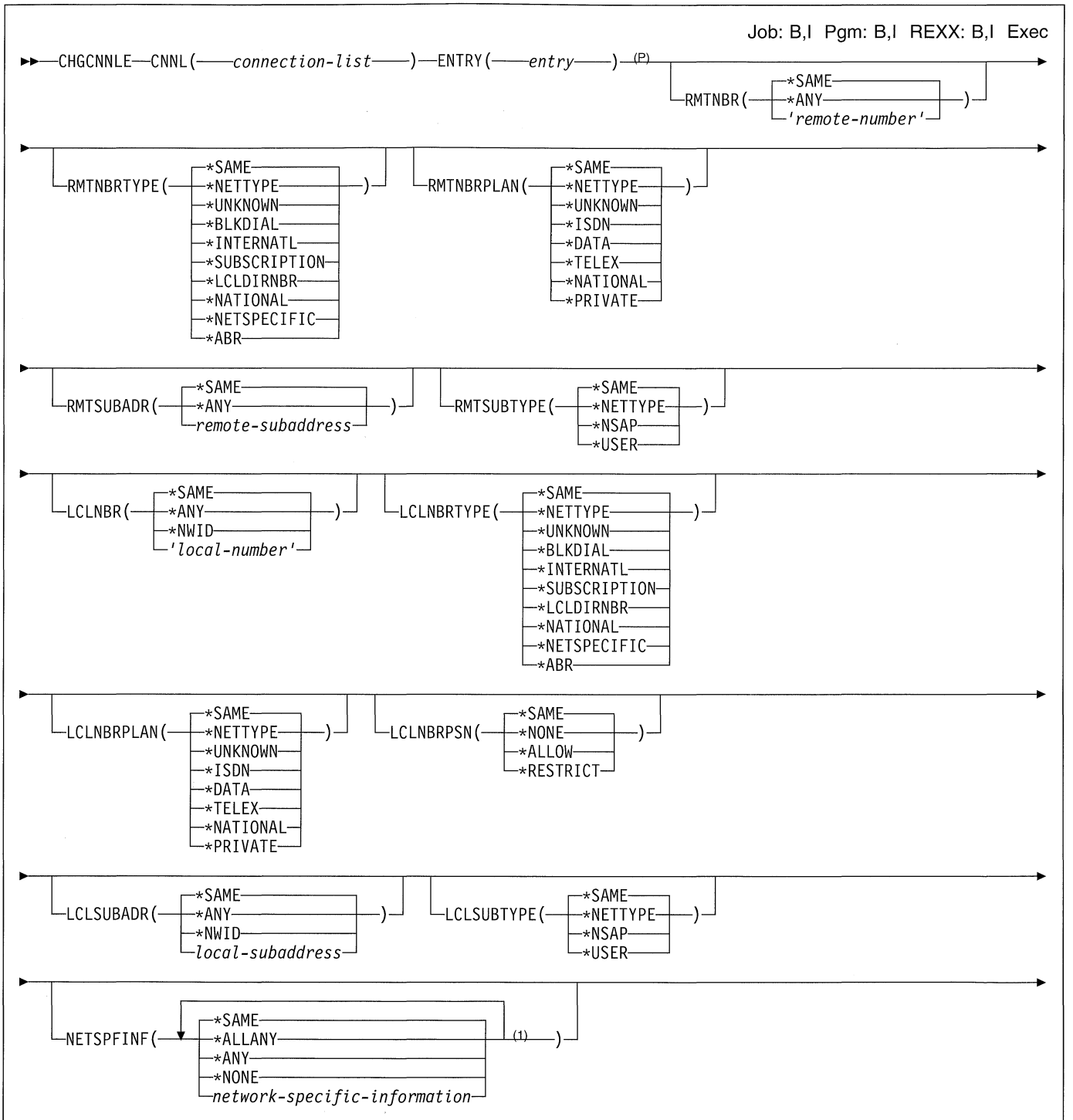
'description': Specify up to 50 characters of text, enclosed in apostrophes.

Example

```
CHGCNNL  CNNL(CHICAGO)
  NETTYPE(*CCITT88)
```

This command changes the network type for connection list CHICAGO to *CCITT88

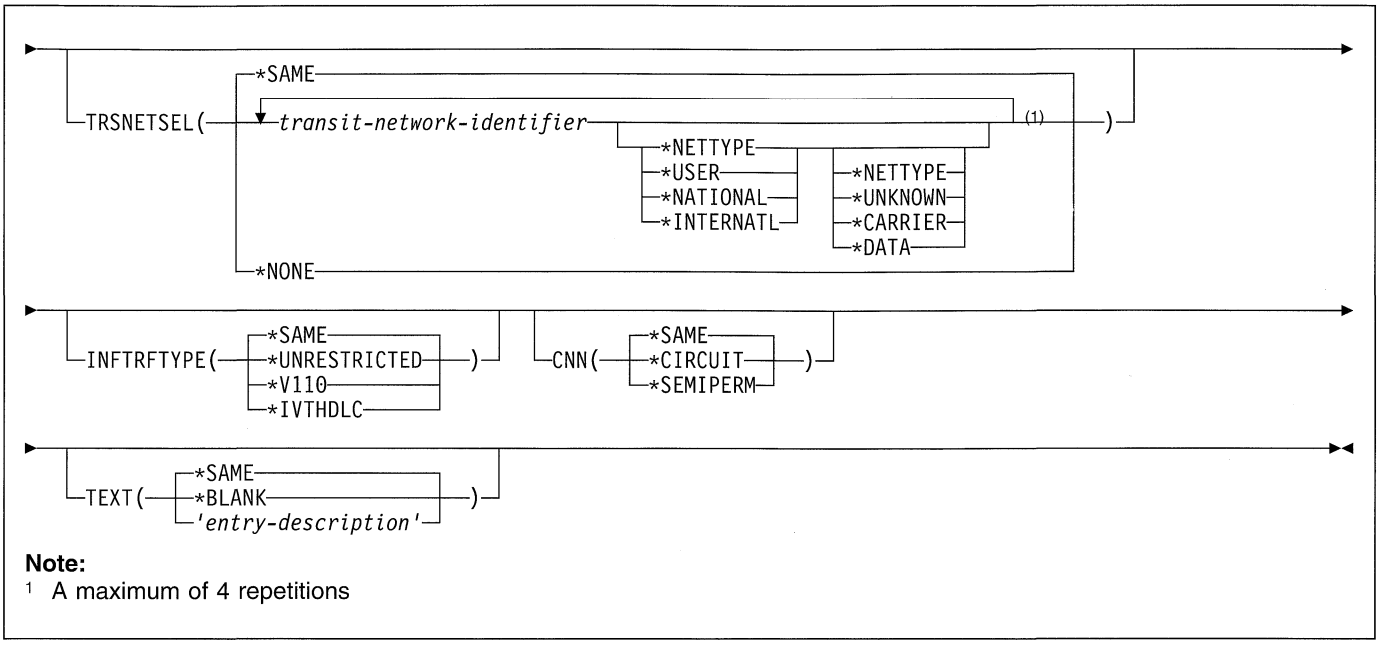
CHGCNNLE (Change Connection List Entry) Command



Notes:

1 A maximum of 4 repetitions

P All parameters preceding this point can be specified in positional form.



Note:
1 A maximum of 4 repetitions

Purpose

The Change Connection List Entry (CHGCNNLE) command changes an entry of a connection list.

Required Parameters

CNNL
Specifies the name of the connection list that contains the entry being changed.

ENTRY
Specifies which entry in the connection list is changed.

Optional Parameters

RMTNBR
Specifies the number of the remote system in the Integrated Services Digital Network (ISDN).

***SAME:** The value does not change.

***ANY:** Any value, including no value, specified in the received Called Party Number information element (IE) (encoded on the call by the system) is acceptable for incoming calls. For outgoing calls, the system requires the target number to be supplied so that a Called Party Number or Keypad Facility IE can be encoded on the outgoing call. Therefore, if *ANY is specified for an outgoing call, the call out attempt fails.

remote-number: Specifies the remote number. Extra characters, such as parentheses, can be used if they are specified on the remove character (RMVCHR) parameter of the connection list.

RMTNBRTYPE
Specifies the type of remote number.

***SAME:** The value does not change.

***NETTYPE:** A default value is used based on the network type specified on the NETTYPE parameter when the connection list was created using the Create Connection List (CRTCNNL) command.

***UNKNOWN:** The remote number type is unknown.

***BLKDIAL:** The remote number type is unknown. *BLKDIAL is used in France instead of *UNKNOWN.

***INTERNATL:** The remote number is an international number type.

***SUBSCRIPTION:** The remote number is a subscription number type.

***LCLDIRNBR:** The remote number is a subscription number type. *LCLDIRNBR is used by AT&T instead of *SUBSCRIPTION.

***NATIONAL:** The remote number is a national address type.

***NETSPECIFIC:** The remote number type is specific to the network.

***ABR:** The remote number type is an abbreviated type.

RMTNBRPLAN
Specifies the numbering plan used for the remote number.

***SAME:** The value does not change.

***NETTYPE:** A default value is used based on the network type specified on the NETTYPE parameter when the connection list was created using the CRTCNNL command.

***UNKNOWN:** The numbering plan is not known.

***ISDN:** The ISDN/telephony numbering plan is used.

***DATA:** The data numbering plan is used.

***TELEX:** The telex numbering plan is used.

***NATIONAL:** The national numbering plan is used.

***PRIVATE:** The private numbering plan is used.

RMTSUBADR
Specifies the subaddress of the remote system.

***SAME:** The value does not change.

***ANY:** Any value, including none, specified in the received Called Party Subaddress IE is acceptable for incoming calls. For outgoing calls, no Called Party Subaddress IE is encoded.

remote-subaddress: Specify the subaddress of the remote system; up to 40 hex characters.

RMTSUBTYPE
Specifies the remote subaddress type.

***SAME:** The value does not change.

***NETTYPE:** A default value is used based on the network type specified on the NETTYPE parameter when the connection list was created using the CRTCNNL command.

***NSAP:** The remote subaddress type is NSAP (X.213).

***USER:** The remote subaddress type is user-specified.

LCLNBR
Specifies information about the local number that is called for an incoming call. If the entry is used for an outgoing call, this parameter is ignored.

***SAME:** The value does not change.

***ANY:** Any value, including no value, specified in the received Called Party Number IE is acceptable.

***NWID:** Any value, including no value, specified in the received Called Party Number IE is acceptable. The number used for outgoing calls is determined by the network interface description.

local-number: Specify the local number. For this entry, only calls directed to this local number are accepted. Extra characters, such as parentheses, can be used if they are specified on the remove character (RMVCHR) parameter of the connection list.

LCLNBRTYPE
Specifies the type of local number given.

***SAME:** The value does not change.

***NETTYPE:** A default value is used based on the network type specified on the NETTYPE parameter when the connection list was created using the CRTCNNL command.

***UNKNOWN:** The local number type is unknown.

***BLKDIAL:** The local number type is unknown. *BLKDIAL is used in France instead of *UNKNOWN.

***INTERNATL:** The local number is an international number type.

***SUBSCRIPTION:** The local number is a subscription number type.

***LCLDIRNBR:** The local number is a subscription number type. *LCLDIRNBR is used by AT&T instead of *SUBSCRIPTION.

***NATIONAL:** The local number is a national address type.

***NETSPECIFIC:** The local number type is specific to the network.

***ABR:** The local number type is an abbreviated type.

LCLNBRPLAN
Specifies the numbering plan used for the local number.

***SAME:** The value does not change.

***NETTYPE:** A default value is used based on the network type specified on the NETTYPE parameter when the connection list was created using the CRTCNNL command.

***UNKNOWN:** The numbering plan is not known.

***ISDN:** The ISDN/telephony numbering plan is used.

***DATA:** The data numbering plan is used.

***TELEX:** The telex numbering plan is used.

***NATIONAL:** The national numbering plan is used.

***PRIVATE:** A private numbering plan is used.

LCLNBRPSN
Specifies the intention of the calling user for the presentation of the local number to the called user. This parameter applies only to outgoing calls.

***SAME:** The value does not change.

***NONE:** The local number presentation is not encoded. The network determines whether to present the local number to the called user.

***ALLOW:** The local number is presented to the called user.

***RESTRICT:** The presentation of the local number to the called user is restricted by the network.

LCLSUBADR
Specifies the subaddress of the local system.

***SAME:** The value does not change.

***ANY:** Any value, including no value, specified in the received Called Party Subaddress IE is acceptable for incoming calls. For outgoing calls, no Called Party Subaddress IE is encoded.

***NWID:** Any value, including no value, specified in the Received Calling Party Subaddress IE is acceptable for incoming calls. For outgoing calls, the Called Party Subaddress IE is encoded on the network interface description.

local-subaddress: Specify the local subaddress; up to 40 hex characters.

LCLSUBTYPE
Specifies the type of subaddress used by the local system.

***SAME:** The value does not change.

***NETTYPE:** A default value is used for the parameter based on the network type specified on the NETTYPE parameter when the connection list was created using the CRTCNNL command.

***NSAP:** The subaddress type is NSAP (X.213).

***USER:** The subaddress type is user specified.

NETSPFINF

Specifies network-specific information.

Network-Specific Information requests special network facilities on an outgoing call or for the network to inform the system of special facilities used on incoming calls. These special facilities and the encoding of the Network-Specific Information are network-specific. If the user wants to use the Network-Specific Information, contact the network provider for information on how to encode it. Some networks can process outgoing calls with this information, but fail to deliver the calls to the remote user. Contact user's network provider to determine if this is the case.

The user can enter up to four network-specific information requests. Requests are processed in chronological order: values of *NONE are automatically moved to the last network-specific information values. The order of the network-specific information values in the outgoing call at the local system must match the order expected at the remote system for incoming calls.

***SAME:** The value does not change.

***ALLANY:** The *ANY value is used for all network-specific information elements. *ALLANY can be specified only once for this parameter.

***ANY:** Any value, including none, specified in the received Network- Specific Facility IE is acceptable for incoming calls. If the entry is used for an outgoing call, no Network-Specific Information is sent.

***NONE:** It is unacceptable to have Network-Specific Information on an incoming call for this entry. If the entry is used for an outgoing call, no Network-Specific Information is sent.

network-specific-information: Specify, in even numbers, the Network-Specific Information up to 60 hex characters. It is encoded into the Network-Specific Facility IE as follows: `x'40' Length<network- specific information)>) <network-specific information>`

TRSNETSEL

Specifies up to four Transit Network Selection IEs. The IE consists of three parts: 1) a transit network identifier, 2) a transit network type, and 3) a transit network plan. The transit network selection IEs are used only when outgoing calls are placed.

If the user wants to use the Transit Network Selection IE, contact the user's network provider for information on how it should be encoded.

The transit network identifier specifies which intermediate ISDN network must be used.

***SAME:** The value does not change.

Element 1: Transit Network Identifier

transit-network-identifier: Specify the 30-character transit network identifier.

Element 2: Transit Network Type

***NETTYPE:** A default value is used based on the network type specified on the NETTYPE parameter when the connection list was created using the CRTCNNL command.

***USER:** The transit network type is user-specified.

***NATIONAL:** The transit network selection IE is a national type.

***INTERNATL:** The transit network selection IE is an international type.

Element 3: Transit Network Plan

***NETTYPE:** A default value is used based on the network type specified on the NETTYPE parameter when the connection list was created using the CRTCNNL command.

***UNKNOWN:** The network plan is unknown.

***CARRIER:** The Carrier identification code plan is used.

***DATA:** The data identification code (X.121) plan is used.

***NONE:** No Transit Network Selection information is sent on the outgoing call.

INFTRFTYPE

Specifies the information transfer type. The information transfer type determines the layer-1 protocol.

***SAME:** The value does not change.

***UNRESTRICTED:** The data-channel traffic appears as digital information; no physical transformation is required and each B-channel operates at capacity (64k bits per second (bps)).

***V110:** The transfer type is V-series Recommendation 110. Each B-channel operates at 56k bps.

***IVTHDLC:** The transfer type is Inverted HDLC. Each B-channel operates at capacity, 64k bps.

CNN

Specifies the type of line connection used.

***SAME:** The value does not change.

***CIRCUIT:** The entry is for a circuit-switched connection.

***SEMIPERM:** The entry is for a semi-permanent connection.

TEXT

Specifies text that briefly describes the entry in the connection list. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

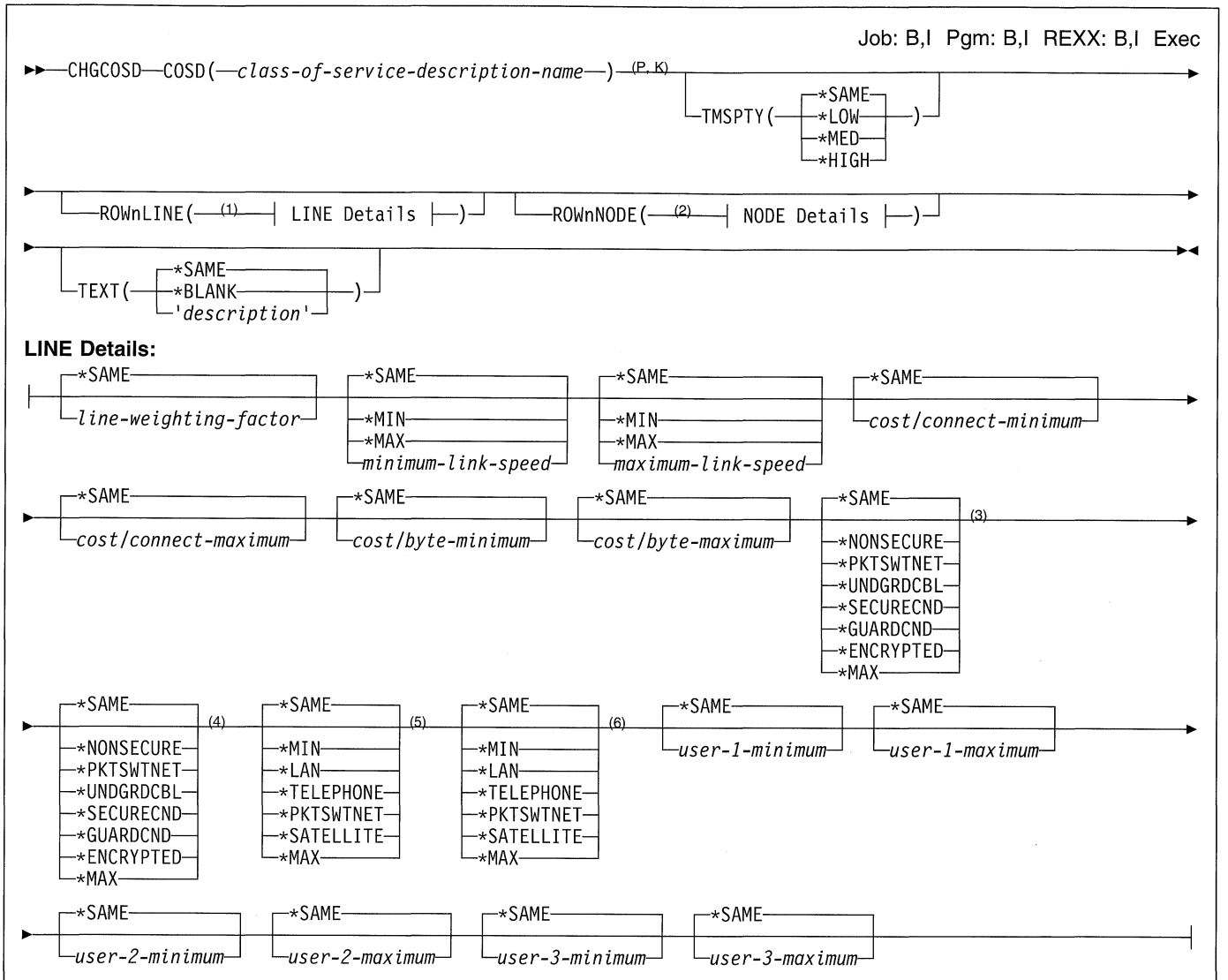
'entry-description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGCNNLE  CNNL(CHICAGO)  ENTRY(CORPORATE)
  RMTNBR('(896) 989-5555')
```

This command changes the remote number to (896) 989-5555 for the CORPORATE entry in the connection list CHICAGO.

CHGCOSD (Change Class-of-Service Description) Command



Notes:

1 n can be a number from 1 to 8. Table 23 shows the defaults for the row line parameters.

2 n can be a value from 1 to 8. Table 24 shows the defaults for the row node parameters.

P All parameters preceding this point can be specified in positional form.

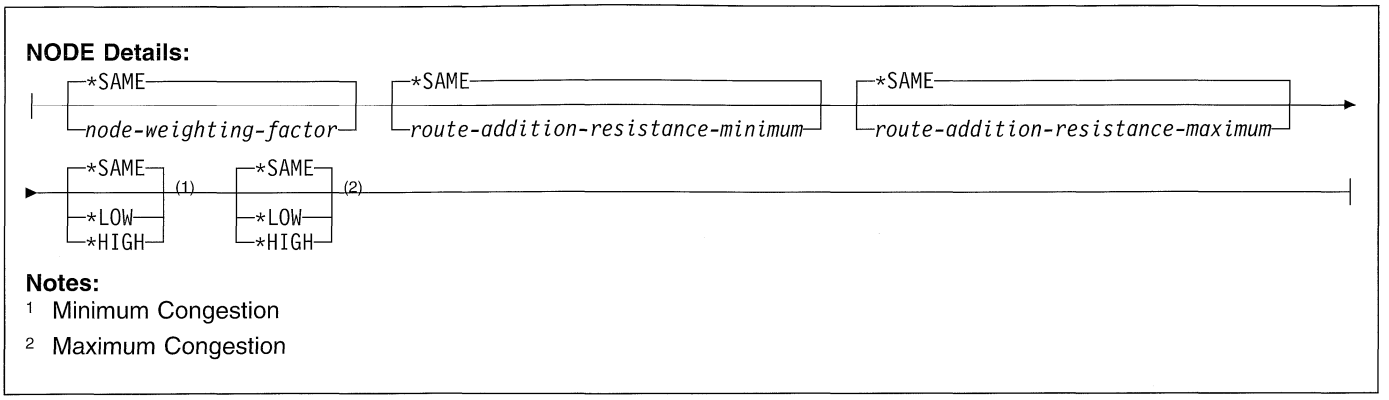
K All parameters preceding this point are key parameters.

3 Minimum Security

4 Maximum Security

5 Minimum Propogation Delay

6 Maximum Propogation Delay



Purpose

The Change Class-of-Service Description (CHGCOSD) command changes a class-of-service description.

Required Parameter

COSD

Specifies the name of the class-of-service description changed. This name ranges from 1 through 8 characters.

Optional Parameters

TMSPTY

Specifies the transmission priority for this class-of-service description. Valid priority levels are listed below.

***SAME:** This value does not change.

***LOW:** The lowest transmission priority for this class-of-service description is used.

***MED:** Medium transmission priority for this class-of-service description is used.

***HIGH:** The highest transmission priority for this class-of-service description is used.

ROW1LINE - ROW8LINE

Specifies the list of line-related criteria used for the first through the eighth rows of the class-of-service description. These rows describe the attributes of the line connection between two nodes in the APPN network. The rows are examined in order (from 1 through 8) to define a network routing path.

Element 1: Line Weight Factor

This is the weighting factor used in the computation.

***SAME:** This value does not change.

line-weighting-factor: Specify the relative weight of this row for lines. The weight, which ranges from 0 through 255, indicates the relative cost of a line connection.

Element 2: Minimum Link Speed

This is the minimum link speed for a line connection that is accepted by this row criteria.

***SAME:** This value does not change.

***MIN:** The minimum link speed is specified.

***MAX:** The maximum link speed is specified.

minimum-link-speed: Specify the minimum link speed. Valid values are *MIN, 1200, 2400, 4800, 7200, 9600, 14400, 19200, 48000, 56000, 64000, 112000, 128000, 168000, 192000, 224000, 256000, 280000, 320000, 336000, 384000, 448000, 499000, 576000, 614000, 691000, 768000, 845000, 922000, 998000, 1075000, 1152000, 1229000, 1382000, 1536000, 1690000, 1843000, 1997000, 4M, 10M, 16M, or *MAX.

Element 3: Maximum Link Speed

This is the maximum link speed for a line connection that is accepted by this row criteria.

***SAME:** This value does not change.

***MIN:** The minimum link speed is specified.

***MAX:** The maximum link speed is specified.

maximum-link-speed: Specify the maximum link speed. See the list of valid values under minimum link speed in Element 2.

Element 4: Minimum Cost Per Connect Time

***SAME:** This value does not change.

cost/connect-minimum: Specify the minimum relative cost per connect time that is accepted by this row criteria. Valid costs range from 0 through 255.

Element 5: Maximum Cost Per Connect Time

***SAME:** This value does not change.

cost/connect-maximum: Specify the maximum relative cost per connect time that is accepted by the row criteria. Valid costs range from 0 through 255.

Element 6: Minimum Cost Per Byte

***SAME:** This value does not change.

cost/byte-minimum: Specify the minimum relative cost per byte that is accepted by this row criteria. Valid costs range from 0 through 255.

Element 7: Maximum Cost Per Byte

***SAME:** This value does not change.

cost/byte-maximum: Specify the maximum relative cost per byte that is accepted by this row criteria. Valid costs range from 0 through 255.

Element 8: Minimum Security Level

The minimum security level that is accepted by this row criteria. Valid values are in order from least to most secure.

***SAME:** This value does not change.

***NONSECURE:** No security is the minimum security level accepted by this row criteria.

***PKTSWTNET:** Packet switched network is the minimum security level accepted by this row criteria.

***UNDGRDCBL:** Underground cable is the minimum security level accepted by this row criteria.

***SECURECND:** Secure conduit is the minimum security level accepted by this row criteria.

***GUARDCND:** Guarded conduit is the minimum security level accepted by this row criteria.

***ENCRYPTED:** Encrypted line is the minimum security level accepted by this row criteria.

***MAX:** Guarded conduit, protected against physical and radiation tapping is the minimum security level accepted by this row criteria.

Element 9: Maximum Security Level

The maximum security level that is accepted by this row criteria. Valid values are in order from least to most secure.

***SAME:** This value does not change.

***NONSECURE:** No security is the maximum security level accepted by this row criteria.

***PKTSWTNET:** Packet switched network is the maximum security level accepted by this row criteria.

***UNDGRDCBL:** Underground cable is the maximum security level accepted by this row criteria.

***SECURECND:** Secure conduit is the maximum security level accepted by this row criteria.

***GUARDCND:** Guarded conduit is the maximum security level accepted by this row criteria.

***ENCRYPTED:** Encrypted line is the maximum security level accepted by this row criteria.

***MAX:** Guarded conduit, protected against physical and radiation tapping is the maximum security level accepted by this row criteria.

Element 10: Minimum Propagation Delay

The minimum propagation delay that is accepted by this row criteria. The valid values are in order from least to longest delay.

***SAME:** This value does not change.

***MIN:** Minimum propagation delay is accepted by this row criteria.

***LAN:** Local Area Network propagation delay is the minimum propagation delay accepted by this row criteria.

***TELEPHONE:** Telephone propagation delay is the minimum propagation delay accepted by this row criteria.

***PKTSWTNET:** Packet switched network propagation delay is the minimum propagation delay accepted by this row criteria.

***SATELLITE:** Satellite propagation delay is the minimum propagation delay accepted by this row criteria.

***MAX:** Maximum propagation delay is the minimum propagation delay accepted by this row criteria.

Element 11: Maximum Propagation Delay

The maximum propagation delay that is accepted by this row criteria. Valid values are in order from least to longest delay.

***SAME:** This value does not change.

***MIN:** Minimum propagation delay is the maximum propagation delay accepted by this row criteria.

***LAN:** Local Area Network propagation delay is the maximum propagation delay accepted by this row criteria.

***TELEPHONE:** Telephone propagation delay is the maximum propagation delay accepted by this row criteria.

***PKTSWTNET:** Packet switched network propagation delay is the maximum propagation delay accepted by this row criteria.

***SATELLITE:** Satellite propagation delay is the maximum propagation delay accepted by this row criteria.

***MAX:** Maximum propagation delay is the maximum propagation delay accepted by this row criteria.

Element 12: User's First Minimum Line Connection Criteria

***SAME:** This value does not change.

user-1-minimum: Specify the user's own line connection criteria. Valid values range from 0 through 255.

Element 13: User's First Maximum Line Connection Criteria

***SAME:** This value does not change.

user-1-maximum: Specify user's own line connection criteria. Valid values range from 0 through 255.

Element 14: User's Second Minimum Line Connection Criteria

***SAME:** This value does not change.

user-2-minimum: Specify user's own line connection criteria. Valid values range from 0 through 255.

Element 15: User's Second Maximum Line Connection Criteria

***SAME:** This value does not change.

user-2-maximum: Specify user's own line connection criteria. Valid values range from 0 through 255.

Element 16: User's Third Minimum Line Connection Criteria

***SAME:** This value does not change.

user-3-minimum: Specify user's own line connection criteria. Valid values range from 0 through 255.

Element 17: User's Third Maximum Line Connection Criteria

***SAME:** This value does not change.

user-3-maximum: Specify user's own line connection criteria. Valid values range from 0 through 255.

ROW1NODE - ROW8NODE

Specifies the list of node-related criteria used for the first through the eighth rows of the class-of-service description. This row describes the attributes of a node in the APPN network. The rows are examined in order (from 1 to 8) to define a network routing path.

Element 1: Node Weighting Factor

***SAME:** This value does not change.

node-weighting-factor: Specify the relative weight of this row for nodes. The weight ranges from 0 through 255.

Element 2: Minimum Route Additional Resistance

***SAME:** This value does not change.

route-addition-resistance-minimum: Specify the minimum route additional resistance accepted by this row criteria. Valid values range from 0 through 255.

Element 3: Maximum Route Additional Resistance

***SAME:** This value does not change.

route-addition-resistance-maximum: Specify the maximum route additional resistance accepted by this row criteria. Valid values range from 0 through 255.

Element 4: Minimum Congestion Level

Specifies the minimum level of congestion tolerated.

***SAME:** This value does not change.

***LOW:** Low congestion is the minimum level tolerated.

***HIGH:** High congestion is the minimum level tolerated.

Element 5: Maximum Congestion Level

Specifies the maximum level of congestion tolerated.

***SAME:** This value does not change.

***LOW:** Low congestion is the maximum level tolerated.

***HIGH:** High congestion is the maximum level tolerated.

TEXT

Specifies text that briefly describes the description. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

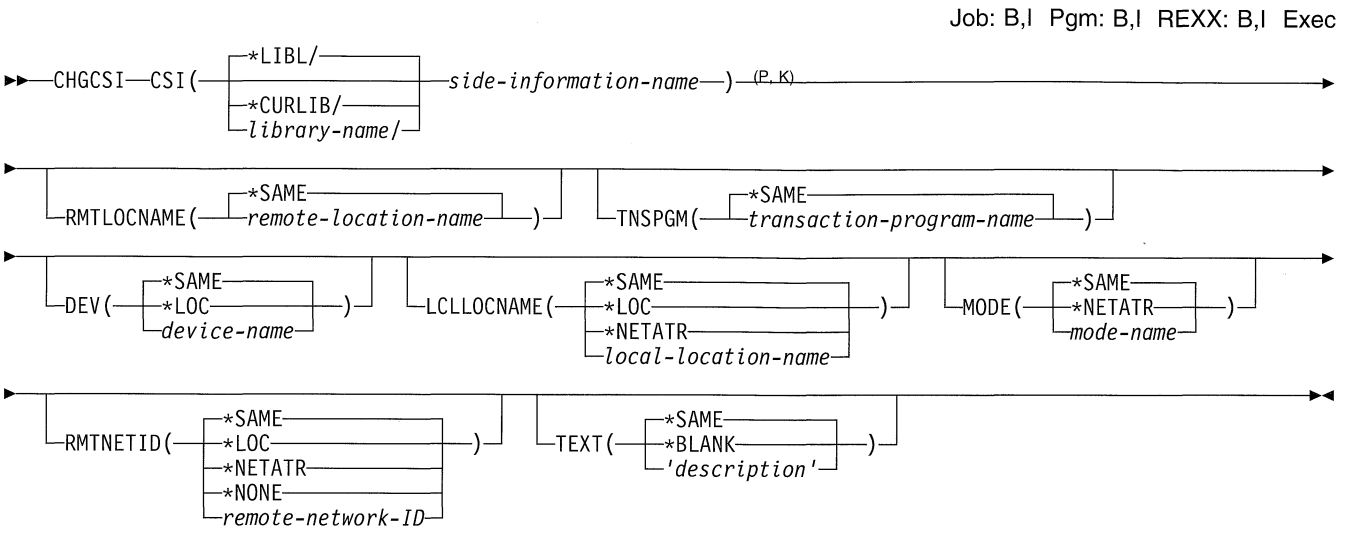
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGCOSD  COSD(COSD1)  ROW4LINE(80 *SAME *SAME 15)
```

This command changes Row 4 line weight to 80 and Row 4 minimum cost/connect time to 15 for class-of-service description COSD1.

CHGCSI (Change Communications Side Information) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Communications Side Information (CHGCSI) command is used to change a side information object in a particular library.

Refer to the *APPC Programmer's Guide* for information on how the system uses the RMTLOCNAME, DEV, LCLLOCNAME, and RMTNETID parameters to select an APPC device description.

Required Parameter

CSI

Specifies the name of the object that contains the desired side information object. The side information object is accessed by the symbolic destination name (sym_dest_name) parameter on the Initialize_Conversation (CMINIT) call.

The name of the side information object can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

side-information-name: Specify the name of the object that contains the side information. This is the CPI-Communications symbolic destination name (sym_dest_name).

Optional Parameters

RMTLOCNAME

Specifies the remote location name of the system with which this object communicates.

The Common Programming Interface (CPI) -Communications partner_LU_name, which consists of the remote network identifier and the remote location, determines the remote logical unit of the remote program.

***SAME:** The value does not change.

remote-location-name: Specify the remote location that is associated with the symbolic destination name.

TNSPGM

Specifies the name (up to 64 characters) of the transaction program on the remote system to be started. This is the CPI-Communications TP_name.

***SAME:** The value does not change.

transaction-program-name: Specify the transaction program name. The user can enter SNA service transaction program names by specifying hexadecimal input. For example, '03F0F0F4' is a valid input format for an SNA service transaction program.

DEV

Specifies the name of the device description used for the remote system.

***SAME:** The value does not change.

***LOC:** The device associated with the remote location is used. If several devices are associated with the remote location, the system determines which device is used.

device-name: Specify the name of a communications device associated with the remote location. If the device name is not valid for the remote location, a message is sent when the program device entry is acquired. More information on device names is in the *APPC Programmer's Guide*.

LCLLOCNAME

Specifies the local location name.

***SAME:** The value does not change.

***LOC:** The device associated with the remote location is used. If several devices are associated with the remote location, the system determines which device is used.

***NETATR:** The LCLLOCNAME value specified in the system network attributes is used.

local-location-name: Specify the name of the local location. Specify the local location if the remote location will also be indicated by a specific local location name.

MODE

Specifies the mode used to control the session. This name is as the Common Programming Interface (CPI)-Communications mode_name.

***SAME:** The value does not change.

***NETATR:** The mode name specified in the network attributes is used.

mode-name: Specify a mode_name for the remote location.

Note: The values SNASVCMG and CPSVCMG cannot be specified.

RMTNETID

Specifies the remote network identifier used with the remote location. The Common Programming Interface (CPI)-Communications partner_LU_name, which consists of the remote network identifier and the remote location, locates the logical unit at which the remote program is located.

***SAME:** The value does not change.

***LOC:** The remote network ID for the remote location is used.

***NETATR:** The RMTNETID value specified in the system network attributes is used.

***NONE:** The remote network has no name.

remote-network-ID: Specify a remote network ID.

TEXT

Specifies text that briefly describes the object. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

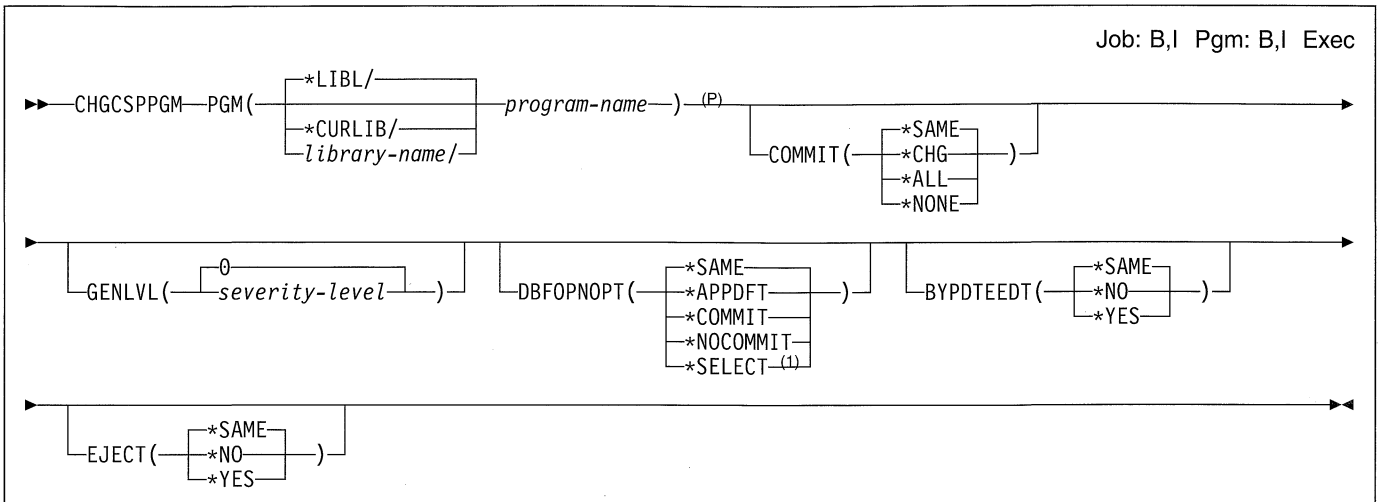
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGCSI CSI(QGPL/SIDEOBJ) RMTNETID(*NETATR)
```

This command changes the remote network ID to *NETATR in the communications side information object SIDEOBJ in library QGPL.

CHGCSPPGM (Change CSP/AE Program) Command



Notes:

P All parameters preceding this point can be specified in positional form.

1 *SELECT not valid in batch.

Purpose

The Change CSP/AE Program (CHGCSPPGM) command changes a Cross System Product/Application Execution (CSP/AE) application program. To ensure program integrity, other jobs running an application acquire a lock preventing this command from changing an application in use.

CHGCSPPGM first checks to ensure that the specified program exists, that it can be allocated for change, and that the current job is authorized to change it. The object must be a CSP/AE application program identified by the attribute CSPAE.

If a parameter value is specified for GENLVL and the specified CSP/AE application program object uses SQL, the SQL statements in the program object are prepared. All SQL statement preparation for normal, output, and error processing actions still applies. If SQL preparation errors exceed the specified GENLVL, then the program is not changed and the escape message CAE9094 is sent to the previous call level of this command; otherwise, the completion message CAE9093 is sent.

Restriction: The user of this command must have object management authority and *CHANGE authority for the program.

Required Parameter

PGM

Specifies the qualified name of the CSP/AE object being changed.

The name of the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the program being changed.

Optional Parameters

COMMIT

Specifies whether SQL statements in the application program are run under commitment control. This parameter may affect files referred to by non-SQL processes. A value other than *NONE causes the SQL/400 program to start a commitment control cycle automatically.

***SAME:** The value does not change.

***CHG:** Only the updated, deleted, and inserted rows are locked until the end of the transaction.

***ALL:** All rows selected, updated, deleted, and inserted are locked until the end of the transaction.

***NONE:** Commitment control is not used. If SQL data definition language (DDL) statements are included in the program, *NONE must be used.

Note: If *CHG and *ALL are specified, DDL statements cannot be included in the application. These SQL/400* DDL statements are:

COMMENT ON
CREATE DATABASE

CREATE INDEX
CREATE TABLE
CREATE VIEW
DROP
GRANT
LABEL ON
REVOKE

GENLVL

Specifies the severity level of errors at which the program is not changed. If, while preparing SQL statements in the program, errors occur with a severity level equal to or greater than the value specified on this parameter, the program is not changed.

0: The default value of 0 (zero) is used.

Some suggested values are:

0 The default severity level.
10 The level value for warnings.
20 The level value for general error messages.
30 The level value for serious error messages.
40 The level value for system-detected error messages.

severity-level: Specify a value ranging from 0 through 40.

DBFOPNOPT

Specifies whether files used by non-SQL processes are opened with commitment control.

***SAME:** The value does not change.

***APPDFT:** Files used by non-SQL processes are opened with commitment control if the following conditions are true:

- A commitment control cycle is active when the file must be opened.
- The file is to be opened for writing.
- The COMMIT parameter value is *CHG or *ALL.
- The application uses explicit commitment control logic.

***COMMIT:** All files used by non-SQL processes are opened with commitment control.

***NOCOMMIT:** All files used by non-SQL processes are opened without commitment control.

***SELECT:** The Open Database Files display is shown. It lists every non-SQL file used by the application. The database file open option can be changed individually for every file using this display.

BYPDTEEDT

Specifies whether data edit checks occur for date fields on maps if the field is filled with blanks or erased with the Field Exit key.

***SAME:** The value does not change.

***NO:** Data edit checks are not bypassed when a date field is cleared. CSP/AE produces an error message stating that an incorrect date format was typed in the field.

***YES:** Data edit checks are bypassed when a date field is cleared. The blank date field is accepted by CSP/AE. The map variable receives a zero value.

EJECT

Specifies whether the first floating print map in a CSP/AE application is printed at the top line of the floating area or on the next available line in the floating area.

***SAME:** The value does not change.

***YES:** The first floating print map in the application is printed at the top of the floating area.

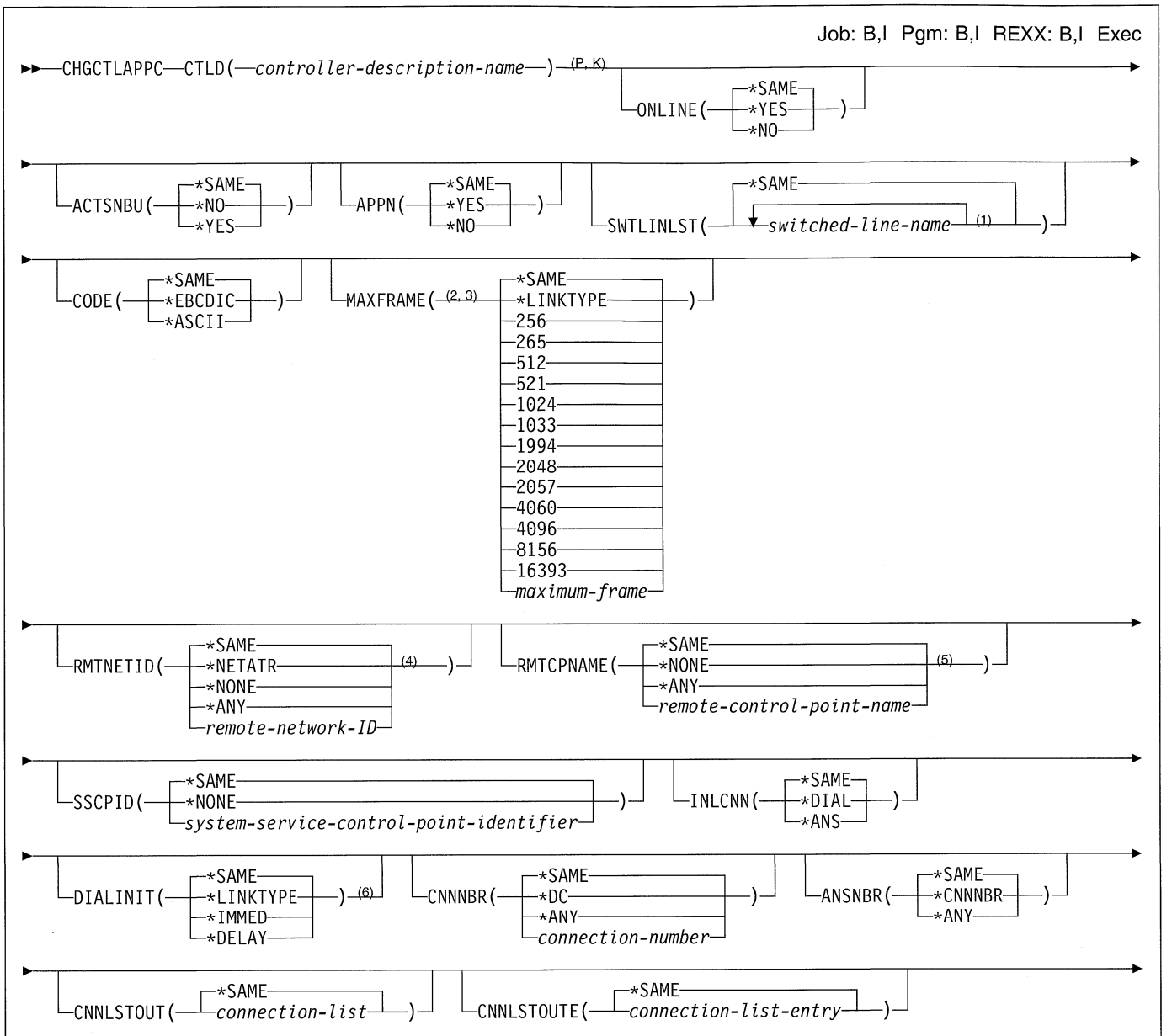
*NO: The first floating print map in the application is printed at the next available floating area line.

Example

```
CHGCSPPGM  PGM(PRODUCTION/INVENT)  COMMIT(*CHG)
  GENLVL(10)  DBFOPNOPT(*NOCOMMIT)
```

This command changes the CSP/AE program INVENT in the library PRODUCTION. If the program uses SQL statements, the statements are prepared using a COMMIT value of *CHG. If SQL preparation errors exceed the severity level of 10, the program does not change. All files used by non-SQL processes will be opened without commitment control.

CHGCTLAPPC (Change Controller Description (APPC)) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

1 A maximum of 64 repetitions

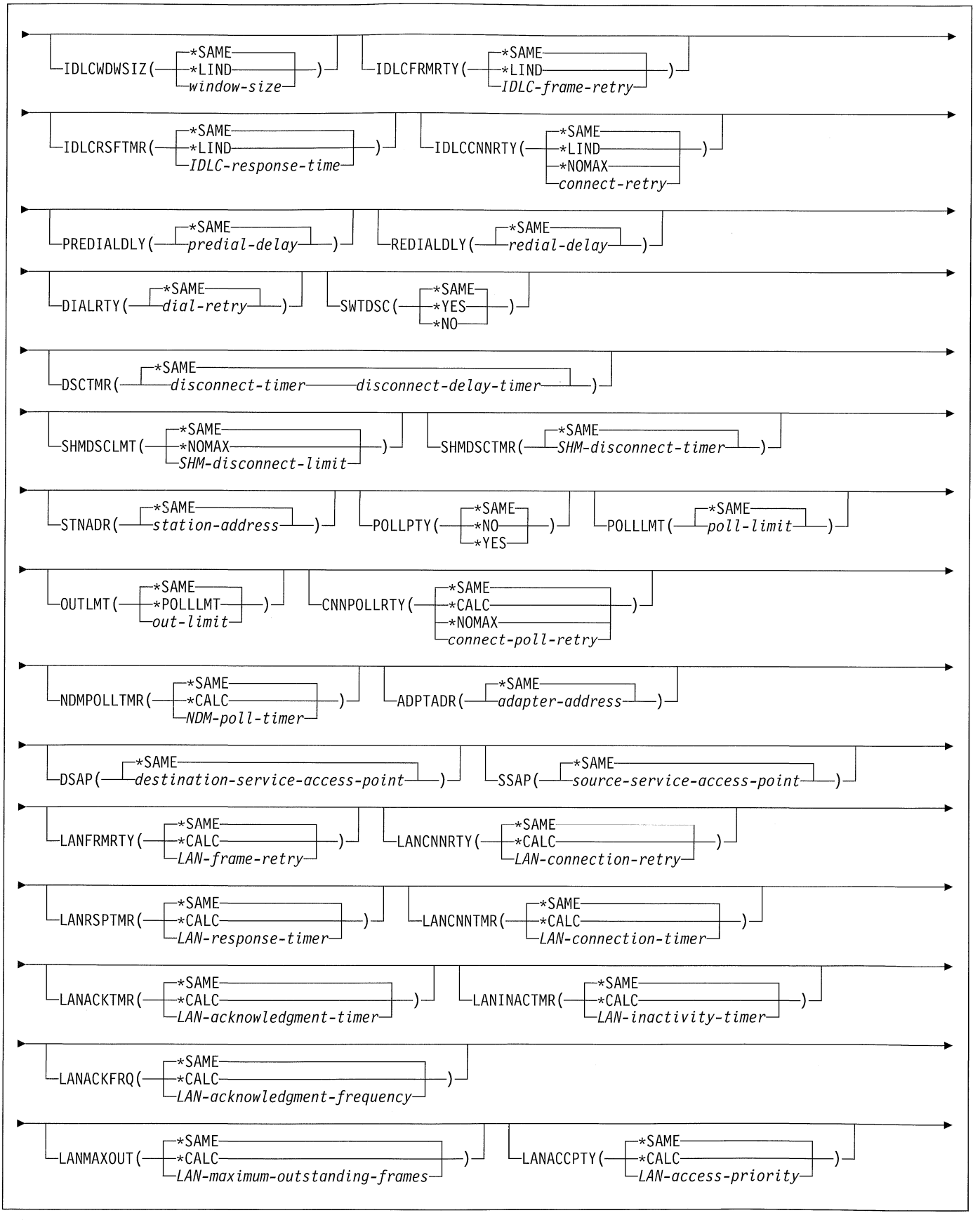
2 If the LINKTYPE is *LAN or *TRLAN, valid values ranging from 265 through 16393.

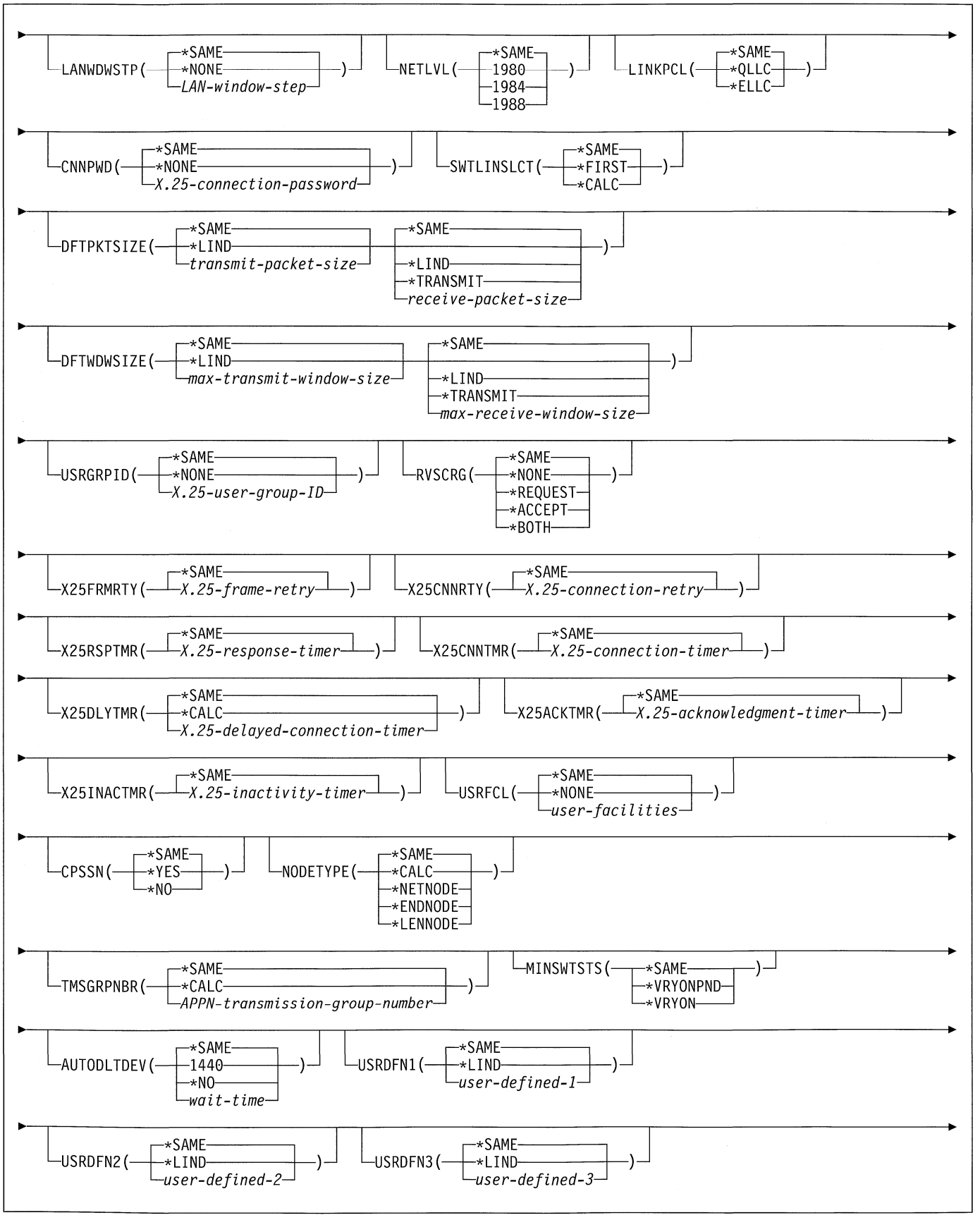
3 For DDI LANs, valid values range from 265 through 4444.

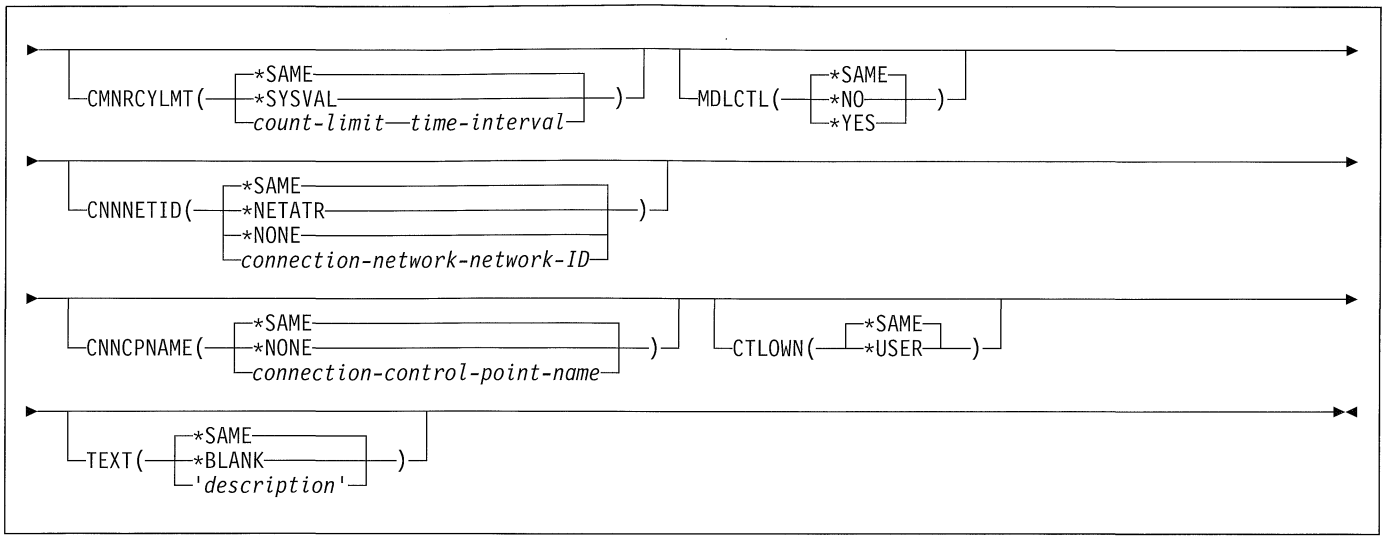
4 If RMTNETID(*ANY) is specified, RMTCPNAME(*ANY) must also be specified.

5 RMTCPNAME(*ANY) cannot be specified if LINKTYPE is *IDLC and SWITCHED or SNBU is *YES; if LINKTYPE is *SDLC and SWITCHED or SNBU is *YES; if NODETYPE is *LENNODE; or if MDLCTL is *YES.

6 This parameter is valid only if INLCNN(*DIAL) is specified.


IDLCWDWSIZ(*SAME *LIND window-size)
IDLCFRMRTY(*SAME *LIND IDLC-frame-retry)
IDLCRSFTMR(*SAME *LIND IDLC-response-time)
IDLCCNNRTY(*SAME *LIND *NOMAX connect-retry)
PREDIALDLY(*SAME predial-delay)
REDIALDLY(*SAME redial-delay)
DIALRTY(*SAME dial-retry)
SWTDSC(*SAME *YES *NO)
DSCTMR(*SAME disconnect-timer disconnect-delay-timer)
SHMDSCLMT(*SAME *NOMAX SHM-disconnect-limit)
SHMDSCTMR(*SAME SHM-disconnect-timer)
STNADR(*SAME station-address)
POLLPTY(*SAME *NO *YES)
POLLLMT(*SAME poll-limit)
OUTLMT(*SAME *POLLLMT out-limit)
CNNPOLLRTY(*SAME *CALC *NOMAX connect-poll-retry)
NDMPOLLTMR(*SAME *CALC NDM-poll-timer)
ADPTADR(*SAME adapter-address)
DSAP(*SAME destination-service-access-point)
SSAP(*SAME source-service-access-point)
LANFRMRTY(*SAME *CALC LAN-frame-retry)
LANCNNRTY(*SAME *CALC LAN-connection-retry)
LANRSPTMR(*SAME *CALC LAN-response-timer)
LANCNNTMR(*SAME *CALC LAN-connection-timer)
LANACKTMR(*SAME *CALC LAN-acknowledgment-timer)
LANINACTMR(*SAME *CALC LAN-inactivity-timer)
LANACKFRQ(*SAME *CALC LAN-acknowledgment-frequency)
LANMAXOUT(*SAME *CALC LAN-maximum-outstanding-frames)
LANACCPTY(*SAME *CALC LAN-access-priority)


LANWDWSTP(*SAME *NONE LAN-window-step)
NETLVL(*SAME 1980 1984 1988)
LINKPCL(*SAME *QLLC *ELLC)
CNNPWD(*SAME *NONE X.25-connection-password)
SWTLINSLCT(*SAME *FIRST *CALC)
DFTPKTSIZE(*SAME *LIND transmit-packet-size *SAME *LIND *TRANSMIT receive-packet-size)
DFTWDWSIZE(*SAME *LIND max-transmit-window-size *SAME *LIND *TRANSMIT max-receive-window-size)
USRGRPID(*SAME *NONE X.25-user-group-ID)
RVSCRG(*SAME *NONE *REQUEST *ACCEPT *BOTH)
X25FRMRTY(*SAME X.25-frame-retry)
X25CNNRTY(*SAME X.25-connection-retry)
X25RSPTMR(*SAME X.25-response-timer)
X25CNNTMR(*SAME X.25-connection-timer)
X25DLYTMR(*SAME *CALC X.25-delayed-connection-timer)
X25ACKTMR(*SAME X.25-acknowledgment-timer)
X25INACTMR(*SAME X.25-inactivity-timer)
USRFCL(*SAME *NONE user-facilities)
CPSSN(*SAME *YES *NO)
NODETYPE(*SAME *CALC *NETNODE *ENDNODE *LENNODE)
TMSGRPNBR(*SAME *CALC APPN-transmission-group-number)
MINSWTSTS(*SAME *VRYONPND *VRYON)
AUTODLTDEV(*SAME 1440 *NO wait-time)
USRDFN1(*SAME *LIND user-defined-1)
USRDFN2(*SAME *LIND user-defined-2)
USRDFN3(*SAME *LIND user-defined-3)



Purpose

The Change Controller Description (APPC) (CHGCTLAPPC) command changes a controller description for an advanced program-to-program communications (APPC) controller.

If the controller was created with LINKTYPE(*LOCAL), the following restrictions apply:

- No communications line can be specified.
- MAXFRAME, RMTNETID, RMTCPNAME, SSCPID, CPSSN, NODETYPE, and TMSGRPNBR must not be specified.
- Parameters relating to SDLC, X.21 short-hold mode, X.25, or LAN configuration must not be specified.

More information is in the *OS/400* Communications Configuration Reference*.

Required Parameter

CTLD
Specifies the name of the controller description being changed.

Optional Parameters

ONLINE
Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The controller is automatically varied on at IPL.

***NO:** This controller is not automatically varied on at IPL.

ACTSNBU
Specifies, for controllers supporting the switched network backup (SNBU) feature, whether the SNBU feature is activated or deactivated. Both the local and remote modems must support the SNBU feature to perform a valid activation.

***SAME:** The value does not change.

***NO:** The switched network backup (SNBU) feature is not activated.

***YES:** The switched network backup feature (SNBU) is activated. To start the feature when the nonswitched connection is broken, specify ACTSNBU(*YES) on the line description command.

APPN
Specifies whether the local system uses advanced peer-to-peer networking (APPN) functions when communicating with this controller. APPC controllers attached to a twinaxial data link control (TDLC) line must have *YES specified.

***SAME:** The value does not change.

***YES:** The local system uses APPN functions.

***NO:** The local system does not use APPN functions.

SWTLINLST
Specifies the names of the switched lines to which this controller attaches. The line descriptions must already exist. Up to 64 switched line names can be specified.

Note: The same line name can be used more than once.

***SAME:** The value does not change.

switched-line-name: Specify the name of a line for which a line description already exists. The maximum number of switched line names that can be connected to the controller is 64.

CODE
Specifies the character code used. The code can be either extended binary-coded decimal interchange code (*EBCDIC) or the American National Standard Code for Information Interchange (*ASCII).

***SAME:** The value does not change.

***EBCDIC:** The extended binary-coded decimal interchange code (EBCDIC) character set code is used.

***ASCII:** The ASCII character set code is used.

MAXFRAME

Specifies the maximum frame (path information unit (PIU)) size that the controller can send or receive. This value is used to calculate request unit (RU) sizes. Since the maximum PIU size that the controller can send or receive is negotiated at exchange identifier time, the maximum PIU size used at run time may be different. This value matches the corresponding value on the host system.

***SAME:** The value does not change.

***LINKTYPE:** The maximum frame size depends on the type of link being used. One of the following values (measured in bytes) will be used if *LINKTYPE is specified:

```
*SDLC-521      *LAN-16393    *TDLC-4105
*IDLC-2048     *X25-1024     *FR  -1590
```

maximum-frame: Specify the maximum frame size for the controller. The frame size that can be used depends on the type of line being used. Valid frame sizes for each line type are shown below.

Link Type	Frame Size (in bytes)
***FR**	265 - 8182
***IDLC**	265 - 8196
***SDLC**	265, 521, 1033, 2057
***X25**	256, 265, 512, 521, 1024, 1033, 2048, 4096
***LAN**	265 - 16393 (265 - 4444 for DDI LANs)
***TDLC**	*LINKTYPE

Note: The numeric values listed for *LINKTYPE are valid only if TYPE(*BLANK) is specified when the controller is created.

RMTNETID

Specifies the name of the remote network in which the adjacent control point resides.

Note: If MDLCTL(*YES) is specified, parameters RMTNETID, RMTCPNAME, and ADPTADR are optional. More information on model controllers is in the *APPN Guide*.

***SAME:** The value does not change.

***NETATR:** The RMTNETID value specified in the system network attributes is used.

***NONE:** No remote network identifier (ID) is used.

***ANY:** The system determines which remote network identifier is used.

remote-network-ID: Specify the remote network identifier.

RMTCPNAME

Specifies the name of the remote control point. This control point name must match the remote system's local control point name. The remote control point name can be seen on the remote system with the DSPNETA command

Note: If MDLCTL(*YES) is specified, parameters RMTNETID, RMTCPNAME, and ADPTADR are optional. More information on model controllers is in the *APPN Guide*.

***SAME:** The value does not change.

***NONE:** No remote control point name is used.

***ANY:** The system determines the name of the remote control point to use.

remote-control-point-name: Specify the remote control point name.

SSCPID

Specifies the system service control point (SSCP) of the host system.

***SAME:** The value does not change.

***NONE:** No system service control point (SSCP) identifier is specified.

system-service-control-point-identifier: Specify the SSCP identifier ranging from 000000000001 through FFFFFFFFFFFF.

INLCNN

Specifies the method used to establish a connection with this controller.

***SAME:** The value does not change.

***DIAL:** The connection is made by a call initiated from the AS/400 system.

For X.25 connection, the line attached to the controller requires switched virtual circuits (SVCs) configured on the LGLCHLE parameter of type OUT or BOTH(*SVCOUT or *SVCBOTH) for the connection to succeed.

***ANS:** The connection is made by the AS/400 system when it answers an incoming call from this controller. If a call is received from the remote controller and all necessary conditions are met, the incoming call is answered by the system.

For X.25 connections, the line to which the controller attaches requires switched virtual circuits (SVC) configured on the LGLCHLE parameter of type IN or BOTH(*SVCIN or *SVCBOTH) for the connection to succeed. The line can be changed using the Change Line Description (X.25) (CHGLINX25) command.

DIALINIT

Specifies the method used to make the initial dial on a switched line between the system and the remote controller.

***SAME:** The value does not change.

***LINKTYPE:** The type of dial connection initiated is specified on the LINKTYPE parameter. For LAN or SDLC short-hold mode connections, the default is to dial the connection immediately upon vary on of the controller description. For all other link types, the default is to delay the dial.

***IMMED:** The dial connection is initiated immediately upon vary on of the controller description.

***DELAY:** The dial connection is delayed until a job is initiated that requests the use of the remote controller resources.

CNNNBR
Specifies the telephone number to dial to connect to this controller.

***SAME:** The value does not change.

***DC:** For X.21 circuit switched connections, a direct call is used to connect to the controller.

***ANY:** Calls are accepted from any network address.

connection-number: Specify the connection number used to call this controller.

This could be a telephone number, an X.25 network address, or an X.21 connection number, depending on the type of controller and the type of line to which it is attached.

ANSNBR
Specifies the X.25 network address from which to accept calls.

***SAME:** The value does not change.

***CNNNBR:** Calls from the X.25 network address specified by the connection number (CNNNBR) parameter are accepted.

***ANY:** Calls are accepted from any X.25 network address.

CNNLSTOUT
Specifies, for ISDN switched connections, the name of a connection list object that contains the Public Switched Network assigned numbers for a dial out operation to the Public Switched Data Network.

***SAME:** The value does not change.

connection-list: Specify the name of a connection list object.

CNNLSTOUTE
Specifies the entry name from the connection list that is used to make a call to the Public Switched Data Network. The connection list must be specified on the CNNLSTOUT parameter.

***SAME:** The value does not change.

connection-list-entry: Specify an entry name.

IDLCWDWSIZ
Specifies the window size used by the line description.

***SAME:** The value does not change.

***LIND:** The window size specified in the line description is used.

window-size: Specify the window size. Valid values range from 1 through 31.

IDLCFRMRTY
Specifies the maximum number of attempts to transmit a frame before an error is reported.

***SAME:** The value does not change.

***LIND:** The number of attempts specified in the line description is used.

IDLC-frame-retry: Specify the number of attempts. Valid values range from 0 through 100.

IDLCRSPTMR
Specifies the amount of time, in tenths of a second, to wait before retransmitting a frame if acknowledgement has not been received.

***SAME:** The value does not change.

***LIND:** The time specified in the line description is used.

IDLC-response-timer: Specify the amount of time to wait before retransmitting an unacknowledged frame of data. Valid values range from 10 through 100 tenths of a second. For example, 10 seconds equals 100 tenths of a second.

IDLCCNNRTY
Specifies the number of times to attempt retransmission at connection time.

***SAME:** The value does not change.

***LIND:** The number of attempts specified in the line description is used.

***NOMAX:** There is no disconnect limit.

IDLC-connect-retry: Specify the number of attempts. Valid values range from 1 through 100.

PREDIALDLY
Specifies how long to wait (in 0.5 second intervals) before dialing.

Note: Predial delay can be specified only if *YES is specified on the SWITCHED or SNBU parameters, *SDLC is specified on the LINKTYPE parameter, and *NO is specified on the SHM parameter.

***SAME:** The value does not change.

predial-delay: Specify a value ranging from 1 through 254 in 0.5 second intervals, or specify 0 to indicate no delay.

REDIALDLY
Specifies how long to wait (in 0.5 second intervals) before re-dialing when the call attempt is unsuccessful. This parameter can be specified only if *YES is specified on either the SWITCHED or SNBU parameter, and if both LINKTYPE(*SDLC) and SHM(*NO) are specified.

***SAME:** The value does not change.

redial-delay: Specify a value ranging from 1 through 254 in 0.5 second intervals, or specify 0 to indicate no delay.

DIALRTY

Specifies the number of re-dial attempts made by the system before considering the dialing unsuccessful.

Note: Dial retries can only be specified if *YES is specified on the SWITCHED or SNBU parameters, *SDLC is specified on the LINKTYPE parameter, and *NO is specified on the SHM parameter.

***SAME:** The value does not change.

dial-retry: Specify a value ranging from 0 through 254 for the number of redial attempts.

SWTDSC

Specifies whether the switched connection to this controller is disconnected when the last session is unbound.

***SAME:** The value does not change.

***YES:** The switched connection is disconnected when the last device is varied off.

***NO:** The switched connection is not disconnected if the last session is unbound.

DSCTMR

Specifies options for controlling the time (in seconds) before an inactive connection is dropped (Element 1), or the amount of time to delay the automatic disconnection (Element 2). If the user does not want the line to disconnect, specify *NO for the switched disconnect prompt.

Element 1: Minimum Connect Timer

***SAME:** The value does not change.

disconnect-timer: Specify a time to wait before disconnecting. Valid values range from 0 through 65535 seconds.

Element 2: Disconnect Delay Timer

***SAME:** The value does not change.

disconnect-delay-timer: Specify a value to delay link take down after the last session on the controller is terminated. Valid values range from 0 through 65535 seconds.

SHMDSCLMT

Specifies the number of nonproductive responses (RR or RNR) that are required from the remote station before the connection can be suspended for this X.21 short-hold mode connection. This parameter is used only if SHM(*YES) is specified.

***SAME:** The value does not change.

***NOMAX:** There is no disconnect limit.

SHM-disconnect-limit: Specify a number ranging from 1 through 254, that indicates the number of nonproductive responses that must be received before the connection can be suspended.

SHMDSCTMR

Specifies the minimum length of time that the primary system maintains the connection to the remote system for this X.21 short-hold mode controller, in tenths of a second. This parameter is used only if SHM(*YES) is specified.

***SAME:** The value does not change.

SHM-disconnect-timer: Specify a value ranging from 2 through 3000, indicating the minimum length of time, in tenths of a second, that the primary maintains the connection to the remote system.

STNADR

Specifies the station address used when communicating with the controller.

***SAME:** The value does not change.

station-address: Specify a 2-character hexadecimal value ranging from 01 through FE.

For secondary controllers, this is the station address of the remote controller. For primary or negotiable controllers, this is the station address of the local system.

POLLPTY

Specifies whether this controller has priority when polled. This parameter can be specified only if *NO is specified on the SHM parameter.

***SAME:** The value does not change.

***NO:** This controller does not have polling priority.

***YES:** This controller has polling priority.

POLLLMT

Specifies the number of consecutive polls that are issued to the same controller when the poll results in receiving frames. This parameter can be specified only if *NO is specified on the SHM parameter.

***SAME:** The value does not change.

poll-limit: Specify a value ranging from 0 through 4 (0 meaning no consecutive polls are attempted) number of consecutive polls.

OUTLMT

Specifies the number of times SDLC allows the consecutive transmission of the maximum number of frames to a station before allowing transmission to another station.

***SAME:** The value does not change.

***POLLLMT:** The value specified on the POLLLMT parameter is used.

out-limit: Specify a value ranging from 0 through 4 for the number of consecutive transmissions.

CNNPOLLRTY

Specifies the number of times to retry connecting before reporting the error. This parameter can be specified only if *NO is specified on the SHM parameter.

***SAME:** The value does not change.

***CALC:** The number of retries is 7 if the controller is switched, and *NOMAX if the controller is nonswitched.

***NOMAX:** There is no disconnect limit.

connect-poll-retry: Specify a value ranging from 0 through 65534 of retries.

NDMPOLLTMR
Specifies the minimum interval at which a secondary station should be polled if a poll from the primary station to the secondary station (which is in normal response mode) does not result in receiving an appropriate response. This parameter can be specified only if *NO is specified on the SHM parameter.

***SAME:** The value does not change.

***CALC:** The system determines the value to use.

NDM-poll-timer: Specify a value ranging from 1 through 3000 in 0.1-second intervals, or 0 to indicate no timer.

ADPTADR
Specifies the 12-character adapter address of the remote controller. This is the address to which the system sends data when it communicates with the remote controller. This value can be obtained from the remote controller's configuration record. Valid values range from hex 000000000001 through hex FFFFFFFFFFFF.

***SAME:** The value does not change.

adapter-address: Specify the adapter address.

DSAP
Specifies the logical address of the SNA destination service access point (DSAP) to which this system sends data when it communicates with the remote controller. This address allows the controller and the system to route the data that comes from this system. The value must match the value specified on the source service access point (SSAP) parameter in the remote controller's configuration record.

***SAME:** The value does not change.

destination-service-access-point: Specify a hexadecimal value ranging from 04 through 9C, in increments of 4 (for example, 04, 08, 0C, 10) to represent the destination service access point.

SSAP
Specifies the source service access point. This is the logical address this system uses when it sends data to the remote controller. This address allows the remote controller and the system to route the data that comes from this system. It must match the value assigned to the destination service access point (DSAP) prompt in the remote controller's configuration record.

***SAME:** The value does not change.

source-service-access-point: Specify a hexadecimal value ranging from 04 through 9C, in increments of 4 (for example, 04, 08, 0C, 10) to represent the source service access point.

LANFRMRTY
Specifies the number of times a frame is transmitted if there is no acknowledgment from the remote controller in the time period specified by the LANRSPTMR parameter. This value is used only after a successful connection has been made.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-frame-retry: Specify a value ranging from 0 through 254 for the number of times a frame will be transmitted before and acknowledgement is received.

LANCNNRTY
Specifies the number of times a transmission is attempted before an acknowledgement is received. This value is used at connection time (unlike LANFRMRTY which is used *after* a connection has been made).

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-connection-retry: Specify a value ranging from 0 through 254 for the number of times the transmission is attempted before an acknowledgement is received.

LANRSPTMR
Specifies the length of time the system waits before an inoperative condition occurs on a link after a connection is made.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-response-timer: Specify a value ranging from 1 through 254 in 0.1-second intervals for the amount of time the system waits before an inoperative condition occurs.

LANCNNTMR
Specifies the length of time the system waits before an inoperative condition occurs on a link at connection time.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-connection-timer: Specify a value ranging from 1 through 254 in 0.1-second intervals for the amount of time the system waits before an inoperative condition occurs. The value 0 indicates that a 6-second connection timer is used.

LANACKTMR
Specifies the length of time the system waits before sending an acknowledgement for received frames.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-acknowledgment-timer: Specify a value, ranging from 1 through 254 in 0.1-second intervals, or specify 0 to indicate no delay. If 0 is specified for this parameter,

0 must also be specified for the LANACKFRQ parameter. If a nonzero value is specified for this parameter, a nonzero value must also be specified for the LANACKFRQ parameter.

LANINACTMR
Specifies the length of time used to determine an inactive condition for the controller.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-inactivity-timer: Specify a value ranging from 1 through 255 in 0.1-second intervals for the length of time used to determine an inactive condition for the controller. The value 0 indicates that a 6-second connection time is used.

LANACKFRQ
Specifies the maximum number of frames that is received before an acknowledgment is sent to the controller.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-acknowledgment-frequency: Specify a value ranging from 0 through 127 for the number of frames received. If 0 is specified for this parameter, 0 must also be specified for the LANACKTMR parameter; if a nonzero value is specified for this parameter, a nonzero value must also be specified for the LANACKTMR parameter.

LANMAXOUT
Specifies the maximum number of frames that can be sent before an acknowledgment is received from the remote system.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-maximum-outstanding-frames: Specify a value ranging from 1 through 127 for the number of frames that can be sent before an acknowledgement is received.

LANACCPTY
Specifies the priority granted to the sending system for sending frames. The larger the number, the higher the priority.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-access-priority: Specify a value ranging from 0 through 3 to indicate how soon a frame may be sent on a local area network.

LANWDWSTP
Specifies whether to reduce to 1 the maximum number of frames outstanding to the remote system during network congestion. This parameter indicates the number of frames that must be successfully received by the remote system before the number of maximum outstanding frames can be increased by 1. The increase continues this way until the maximum number of outstanding frames reaches the value specified by the LAN maximum outstanding frames (LANMAXOUT) parameter.

***SAME:** The value does not change.

***NONE:** The number of outstanding frames is not reduced during network congestion.

LAN-window-step: Specify a value from 1 to 127 for the number of frames that must be successfully received by the remote system before the maximum number of outstanding frames can be increased by 1.

NETLVL
Specifies the level of X.25 support implemented by the network and remote DTE. The level is specified by giving the year of the CCITT standard implemented.

Note: It is suggested to use the lower value of the remote DTE or the network level; for example, if the remote DTE is using the CCITT standard of 1980 and the network 1984, specify 1980 for this prompt.

***SAME:** The value does not change.

1980: The 1980 standard is used.

1984: The 1984 standard is used.

1988: The 1988 standard is used.

LINKPCL
Specifies the link level protocol used on the X.25 network to communicate with this controller.

***SAME:** The value does not change.

***QLLC:** The QLLC protocol is used.

***ELLC:** The ELLC protocol is used.

CNNPWD
Specifies, for X.25 switched virtual circuit (SVC) controllers, the password used when connecting to this controller.

***SAME:** The value does not change.

***NONE:** No password is used.

X.25-connection-password: Specify the connection password. This password for each controller can consist of any alphanumeric characters represented by the hexadecimal range from 40 through FF.

SWTLINSLCT
Specifies the method that is used to select lines from an X.25 switched line list.

***SAME:** The value does not change.

***FIRST:** The lines are selected in the order in which they are specified.

***CALC:** The system determines the value to use.

DFTPKTSIZE

Specifies the default packet size used by the X.25 network.

Element 1: Transmit Packet Size

***SAME:** The value does not change.

***LIND:** The value specified in the line description is used as the default packet size.

transmit-packet-size: Specify a default packet size for transmission. The valid values for the packet size are 64, 128, 256, 512, 1024, 2048, and 4096.

Element 2: Receive Packet Size

***SAME:** The value does not change.

***LIND:** The value specified in the line description is used as the default packet size.

***TRANSMIT:** The value specified as the default packet size for transmission is used as the default for reception.

receive-packet-size: Specify a default packet size for reception. The valid values for the packet size are 64, 128, 256, 512, 1024, 2048, and 4096.

DFTWDWSIZE

Specifies the default window size used by the X.25 network.

Element 1: Transmit Window Size

***SAME:** The value does not change.

***LIND:** The value specified in the line description is used as the default window size.

max-transmit-window-size: Specify the appropriate default window size. Valid values range from 1 through 7 for networks that use modulus 8 packet numbering. Valid values range from 1 through 15 for networks that use modulus 128-packet numbering.

Element 2: Receive Window Size

***SAME:** The value does not change.

***LIND:** The value specified in the line description is used as the default window size.

***TRANSMIT:** The value specified as the default window size for transmission is used as the default for reception.

max-receive-window-size: Specify the appropriate default window size. Valid values range from 1 through 7 for networks that use modulus 8 packet numbering. Valid values range from 1 through 15 for networks that use modulus 128-packet numbering.

USRGRPID

Specifies the closed user-group ID for contacting this X.25 switched virtual circuit (SVC) controller. Specify a value ranging from 0 through 99, as provided by the network subscription. This parameter is not valid for permanent virtual circuit (PVC) connections. It is valid only for SVC circuit outgoing call operations and is ignored for SVC incoming call connections.

***SAME:** The value does not change.

***NONE:** No user-group ID is specified.

X.25-user-group-ID: Specify a value ranging from 0 through 99, as provided by the network subscription.

RVSCRG

Specifies whether reverse charges are accepted or requested when contacting this controller.

***SAME:** The value does not change.

***NONE:** No reverse charging for network tariff billing is accepted.

***REQUEST:** Charges are requested on outgoing call request packets.

***ACCEPT:** Reverse charging for network tariff billing is accepted on incoming requests.

***BOTH:** Both incoming and outgoing requests are accepted.

X25FRMRTY

Specifies the maximum number of times that a logical link control (LLC) protocol data unit is sent after the response timer ends when connected to this controller. The value used in this prompt depends on the quality of service provided by the network and the connection to that network; that is, the frequency of lost link protocol data units.

***SAME:** The value does not change.

X.25-frame-retry: Specify a value ranging from 0 through 21 for the number of times a frame is sent.

X25CNNRTY

Specifies, for X.25 controllers specifying *SEC or *NEG as the role parameter, the maximum number of times that an LLC is sent after the connect response timer ends when connecting to this controller.

***SAME:** The value does not change.

X.25-connection-retry: Specify a value ranging from 0 through 21 retries.

X25RSPTMR

Specifies the time allowed to return an acknowledgment when an LLC is sent while connecting to this controller.

***SAME:** The value does not change.

X.25-response-timer: Specify a value ranging from 1 through 2550 in 0.1-second intervals.

X25CNNTMR

Specifies, for X.25 controllers specifying *SEC or *NEG as the role parameter, this specifies the time allowed for acknowledgment to be returned when a LLC is sent while connecting to this controller.

***SAME:** The value does not change.

X.25-connection-timer: Specify a value ranging from 1 through 2550 in 0.1-second intervals.

X25DLYTMR

Specifies, for X.25 controllers specifying *SEC or *NEG as the role parameter, how often to try establishing a connection to the controller.

***SAME:** The value does not change.

***CALC:** The AS/400 system uses the values for X25CNNTMR and X25CNNRTY to determine how often and how many times to try to establish the connection.

X.25-delayed-connection-timer: Specify a value ranging from 1 through 32767 in 0.1-second intervals. The system retries the connection indefinitely at the specified interval.

X25ACKTMR

Specifies, for X.25 links using an ELLC link protocol, the time period to delay sending acknowledgments for received LLC.

***SAME:** The value does not change.

X.25-acknowledgment-timer: Specify a value ranging from 1 through 2550 in 0.1-second intervals, or specify 0 to indicate no delay.

X25INACTMR

Specifies the time period in which to determine an inactive condition for the controller.

***SAME:** The value does not change.

X.25-inactivity-timer: Specify a value ranging from 1 through 2550 in 0.1-second intervals.

USRFCL

Specifies a string of hexadecimal digits sent to the X.25 network to request additional services. The AS/400 system allows up to 218 hexadecimal characters.

***SAME:** The value does not change.

***NONE:** No additional services are needed.

user-facilities: Specify a string of hexadecimal digits sent to the network to request additional services. The AS/400 system allows up to 218 hexadecimal characters.

CPSSN

Specifies whether this controller supports APPN control-point to control-point (CP-to-CP) sessions.

***SAME:** The value does not change.

***YES:** This controller supports CP-to-CP sessions.

***NO:** This controller does not support CP-to-CP sessions.

NODETYPE

Specifies the type of APPN node this controller represents.

***SAME:** The value does not change.

***CALC:** The system determines the value to use.

***NETNODE:** This node is a network node in an APPN network.

***ENDNODE:** This node is an end node in an APPN network.

***LENNODE:** This node is a low entry networking node in an APPN network.

TMSGRPNBR

Specifies the APPN transmission group number for this controller.

***SAME:** The value does not change.

***CALC:** The system determines the value to use.

APPN-transmission-group-number: Specify a value ranging from 1 through 20 for the transmission group number.

MINSWTSTS

Specifies, for the switched connection, the minimum status that APPN requires for a controller to be considered available for routing.

***SAME:** The value does not change.

***VRYONPND:** APPN will consider the controller available for routing if the status is vary on pending, varied on, or active.

***VRYON:** Indicates that APPN will consider the controller available for routing only if the status is varied on or active.

AUTODLTDEV

Specifies the number of minutes an automatically created device can remain when the last session is unbound before the device description is varied off and deleted.

***SAME:** The value does not change.

1440: The system will automatically vary off and delete the automatically-configured idle device descriptions after 1440 minutes (24 hours).

***NO:** The system will not automatically vary off and delete the automatically-configured idle device descriptions.

wait-time: Specify the number of minutes to wait before deleting the automatically-configured idle device descriptions for this controller. Valid values range from 1 through 10,000.

USRDFN1

Specifies the first of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

***LIND:** The user-defined value specified in the line description is used.

user-defined-1: Specify a value ranging from 0 through 255.

USRDFN2

Specifies the second of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

***LIND:** The user-defined value specified in the line description is used.

user-defined-2: Specify a value ranging from 0 through 255.

USRDFN3

Specifies the third of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

***LIND:** The user-defined value specified in the line description is used.

user-defined-3: Specify a value ranging from 0 through 255.

CMNRCYLMT

Specifies the number of recovery attempts made by the system before an inquiry message is sent to the system operator. Also specifies the time (in minutes) that must elapse before the system sends an inquiry message to the system operator indicating that the recovery attempt count limit is reached.

***SAME:** The value does not change.

***SYSVAL:** The recovery limits specified in the QCMNRCYLMT system value are used.

Element 1: Maximum Recovery Limit

count-limit: Specify the number of second-level recovery attempts to be made. Valid values range from 0 through 99.

Element 2: Recovery Time Interval

time-interval: Specify the time interval (in minutes) at which the specified number of second-level recoveries are attempted. Valid values range from 0 through 120. If the value specified for *count-limit* is not 0, the value 0 specifies infinite recovery.

MDLCTL

Specifies whether this controller description is a *model* for automatically created controller descriptions. Values on the model description, such as timer delays, retry limits, and frame size, will be used for new controller descriptions that are automatically created and configured when communications with a remote system is started. The new controller must be attached to one of the SINGLE line descriptions in the switched line list (SWTLINLST parameter) of the model controller.

A model controller description will not be attached to any devices, and only one controller description can be varied on for each line description.

More information on model controllers is in the *APPN Guide*.

Note: This parameter is valid only if the parameter LINKTYPE(*LAN) is specified.

***SAME:** The value does not change.

***NO:** This controller description is not used as a model controller description.

***YES:** This controller description is used as a model for automatically created controller descriptions.

CNNNETID

Specifies the connection network network identifier of this controller description. If a value is specified for this parameter (other than *NONE), this controller description represents this connection to the connection network.

Note: This parameter is only valid if MDLCTL(*YES) is specified.

***SAME:** The value does not change.

***NETATR:** The LCLNETID value specified in the system network attributes is used.

***NONE:** No network network connection identifier is used.

connection-network-network-ID: Specify the network connection identifier that will represent this controller description to the network.

CNNCPNAME

Specifies the connection network control point name for this controller. If this value and a CNNNETID value (other than *NONE) are specified, this controller description represents this connection to the connection network.

Note: This parameter is valid only if MDLCTL(*YES) is specified.

***SAME:** The value does not change.

***NONE:** No connection network control point name is specified.

connection-control-point-name: Specify the connection control point name.

CTLOWN

Specifies whether the ownership of the controller will be transferred from the system to the user. If the user is the current owner, ownership cannot be transferred back to the system.

***SAME:** The value does not change.

***USER:** The user will be the owner of this controller.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

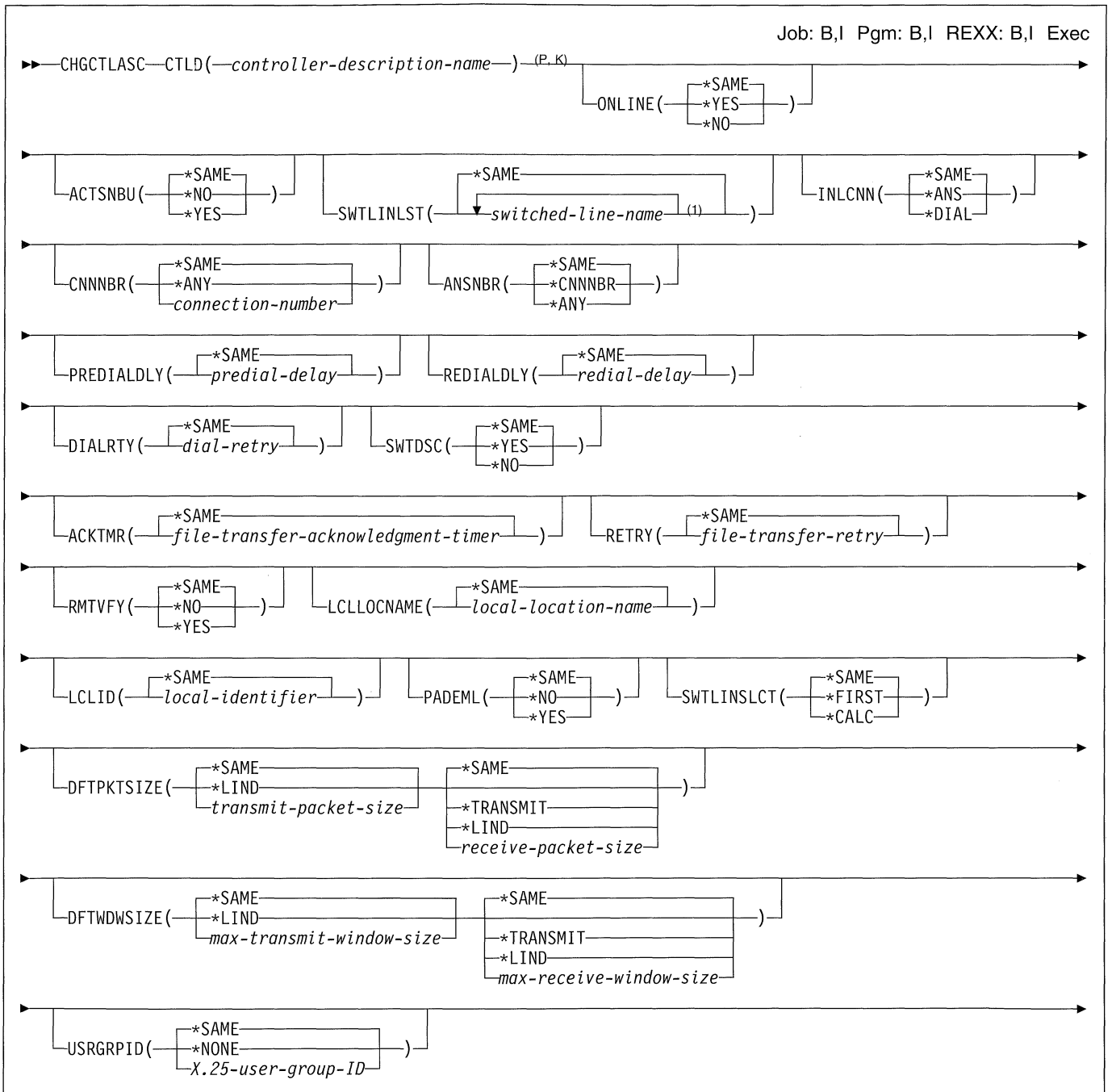
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGCTLAPPC  CTLD(CHICAGO)  ONLINE(*YES)
  TEXT('Controller in Chicago')
```

This command changes the APPC controller named CHICAGO to be varied on at IPL and a new text description is specified.

CHGCTLASC (Change Controller Description (Async)) Command

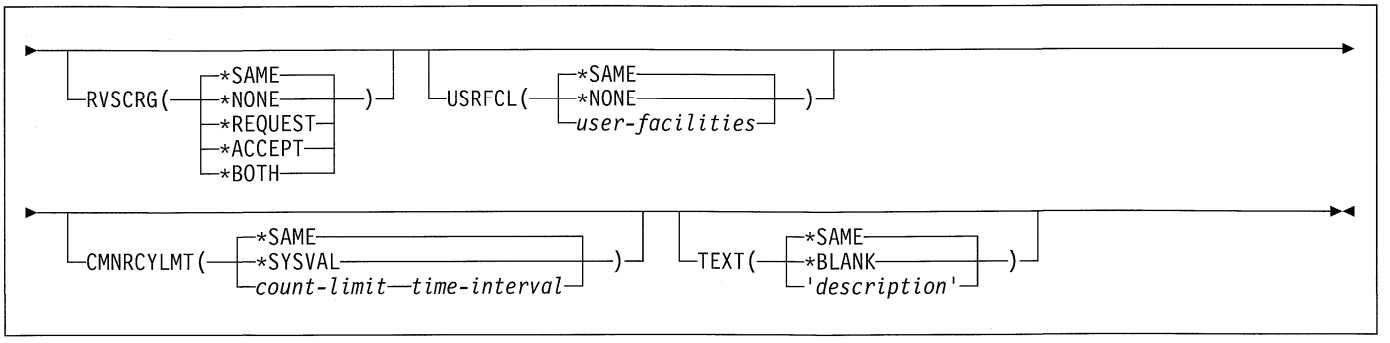


Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

1 A maximum of 64 repetitions



Purpose

The Change Controller Description (Async) (CHGCTLASC) command changes a controller description for an asynchronous controller.

Required Parameter

CTLD
Specifies the name of the controller description being changed.

Optional Parameters

ONLINE
Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** This value does not change.

***YES:** The controller is automatically varied on at IPL.

***NO:** This controller is not automatically varied on at IPL.

ACTSNBU
Specifies, for controllers supporting the switched network backup (SNBU) feature, whether the SNBU feature is activated or deactivated. Both the local and remote modems must support the SNBU feature to perform a valid activation.

***SAME:** The value does not change.

***NO:** The switched network backup (SNBU) feature is not activated.

***YES:** The switched network backup feature (SNBU) is activated. To start the feature when the nonswitched connection is broken, specify ACTSNBU(*YES) on the line description command.

SWTLINLST
Specifies the names of the switched lines to which this controller attaches. The line descriptions must already exist. Up to 64 switched line names can be specified.

***SAME:** The value does not change.

switched-line-name: Specify the names of up to 64 lines that are connected to this controller description. The same line name can be used more than once. For each line name specified, a line description by that name must already exist.

INLCNN
Specifies the method used to establish a connection with this controller.

***SAME:** The value does not change.

***ANS:** The connection is made by the AS/400 system when it answers an incoming call from this controller. If a call is received from the remote controller and all necessary conditions are met, the incoming call is answered by the system.

For X.25 connections, the line to which the controller attaches requires switched virtual circuits (SVC) configured on the LGLCHLE parameter of type IN or BOTH(*SVCIN or *SVCBOTH) for the connection to succeed. The line can be changed using the Change Line Description (X.25) (CHGLINX25) command.

***DIAL:** The connection is made by a call initiated from the AS/400 system.

For X.25 connection, the line attached to the controller requires switched virtual circuits (SVCs) configured on the LGLCHLE parameter of type OUT or BOTH(*SVCOUT or *SVCBOTH) for the connection to succeed.

CNNNBR
Specifies the telephone number to dial to connect to this controller.

***SAME:** The value does not change.

***ANY:** Calls are accepted from any network address.

connection-number: Specify the connection number used to call this controller.

ANSNBR
Specifies the X.25 network address from which to accept calls.

***SAME:** The value does not change.

***CNNNBR:** Calls from the X.25 network address specified by the connection number (CNNNBR) parameter are accepted.

***ANY:** Calls are accepted from any X.25 network address.

PREDIALDLY
Specifies how long to wait (in 0.5 second intervals) before dialing.

***SAME:** The amount of time does not change.

predial-delay: Specify a value ranging from 1 through 254 in 0.5 second intervals, or specify 0 to indicate no delay.

REDIALDLY
Specifies how long to wait (in 0.5 second intervals) before re-dialing when the call attempt is unsuccessful.

***SAME:** The value does not change.

redial-delay: Specify a value ranging from 1 through 254 in 0.5 second intervals, or specify 0 to indicate no delay.

DIALRTY
Specifies the number of re-dial attempts made by the system before considering the dialing unsuccessful.

***SAME:** The value does not change.

dial-retry: Specify a value ranging from 0 through 254 for the number of dial attempts.

SWTDSC
Specifies whether the switched connection to this controller is made inoperative when the last device is varied off.

***SAME:** The value does not change.

***YES:** The switched connection is disconnected when the last device is varied off.

***NO:** The switched connection is not made inoperative when the last device is varied off.

ACKTMR
Specifies the time allowed for an acknowledgment on an asynchronous file transfer.

***SAME:** The value does not change.

file-transfer-acknowledgment-timer: Specify a value ranging from 16 through 65535 seconds to allow for an acknowledgment.

RETRY
Specifies the number of retries on an Async file transfer.

***SAME:** The value does not change.

file-transfer-retry: Specify a value ranging from 1 through 255 for the number of retries.

RMTVFY
Specifies whether the remote system requires verification of local location name and local ID.

***SAME:** The value does not change.

***NO:** The remote system does not require verification of local location name and local ID.

***YES:** The remote system requires verification of local location name and local ID.

LCLLOCNAME
Specifies the local location name.

Note: This name must be the same as that specified by the remote system in its remote location list.

***SAME:** The value does not change.

local-location-name: Specify the local location name.

LCLID
Specifies the ID which, when combined with the local location name, identifies the user's controller to a remote system. This ID must be the same ID specified by the remote system in its remote location list.

***SAME:** The value does not change.

local-identifier: Specify the local identifier.

PADEML
Specifies whether this controller emulates X.25 packet assembly/disassembly (PAD). This PAD emulation follows recommendations for X.3, X.28, and X.29.

***SAME:** The value does not change.

***NO:** This controller does not emulate X.25 packet assembly/disassembly (PAD).

***YES:** This controller emulates X.25 packet assembly/disassembly (PAD).

SWTLINSLCT
Specifies the method that is used to select lines from an X.25 switched line list.

***SAME:** The value does not change.

***FIRST:** The lines are selected in the order in which they are specified.

***CALC:** The system determines the value to use.

DFTPKTSIZE
Specifies the default packet size used by the X.25 network.

Element 1: Transmit Packet Size

***SAME:** The value does not change.

***LIND:** The value specified in the line description is used as the default packet size..

transmit-packet-size: Specify a default packet size for transmission. The valid values for the packet size are 64, 128, 256, 512, 1024, 2048, and 4096.

Element 2: Receive Packet Size

***SAME:** The value does not change.

***TRANSMIT:** The value specified as the default packet size for transmission is used as the default for reception.

***LIND:** The value specified in the line description is used as the default packet size..

receive-packet-size: Specify a default packet size for reception. The valid values for the packet size are 64, 128, 256, 512, 1024, 2048, and 4096.

DFTWDWSIZE
Specifies the default window size used by the X.25 network.

Element 1: Transmit Window Size

***SAME:** The value does not change.

***LIND:** The value specified in the line description is used as the default window size..

max-transmit-window-size: Specify the appropriate default window size. Valid values range from 1 through 7 for networks that use modulus 8 packet numbering. Valid values range from 1 through 15 for networks that use modulus 128 packet numbering.

Element 2: Receive Window Size

***SAME:** The value does not change.

***TRANSMIT:** The value specified as the default window size for transmission is used as the default for reception.

***LIND:** The value specified in the line description is used as the default window size..

max-receive-window-size: Specify the appropriate default window size. Valid values range from 1 through 7 for networks that use modulus 8 packet numbering. Valid values range from 1 through 15 for networks that use modulus 128 packet numbering.

USRGRPID
Specifies the closed user-group ID for contacting this X.25 switched virtual circuit (SVC) controller. Specify a value ranging from 0 through 99, as provided by the network subscription. This parameter is not valid for permanent virtual circuit (PVC) connections. It is valid only for SVC circuit outgoing call operations and is ignored for SVC incoming call connections.

***SAME:** The value does not change.

***NONE:** No user group ID is specified.

X.25-user-group-ID: Specify a value ranging from 0 through 99 as provided by the network subscription.

RVSCRG
Specifies whether reverse charges are accepted or requested when contacting this controller.

***SAME:** The value does not change.

***NONE:** No reverse charging for network tariff billing is accepted.

***REQUEST:** Charges are requested on outgoing call request packets.

***ACCEPT:** Reverse charging for network tariff billing is accepted on incoming requests.

***BOTH:** Both incoming and outgoing requests are accepted.

USRFCL
Specifies a string of hexadecimal digits sent to the X.25 network to request additional services. The AS/400 system allows up to 218 hexadecimal characters.

***SAME:** This value does not change.

***NONE:** No user facilities request is specified.

user-facilities: Specify a string of hexadecimal digits sent to the network to request additional services. The AS/400 system allows up to 218 hexadecimal characters.

CMNRCYLMT
Specifies the number of recovery attempts made by the system before an inquiry message is sent to the system operator. Also specifies the time (in minutes) that must elapse before the system sends an inquiry message to the system operator indicating that the recovery attempt count limit is reached.

***SAME:** The recovery limits do not change.

***SYSVAL:** The recovery limits specified in the QCMNRCYLMT system value are used.

Element 1: Maximum Recovery Limit

count-limit: Specify the number of second-level recovery attempts to be made. Valid values range from 0 through 99.

Element 2: Recovery Time Interval

time-interval: Specify the time interval (in minutes) at which the specified number of second-level recoveries are attempted. Valid values range from 0 through 120. If the value specified for *count-limit* is not 0, the value 0 specifies infinite recovery.

TEXT
Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** This value does not change.

***BLANK:** Text is not specified.

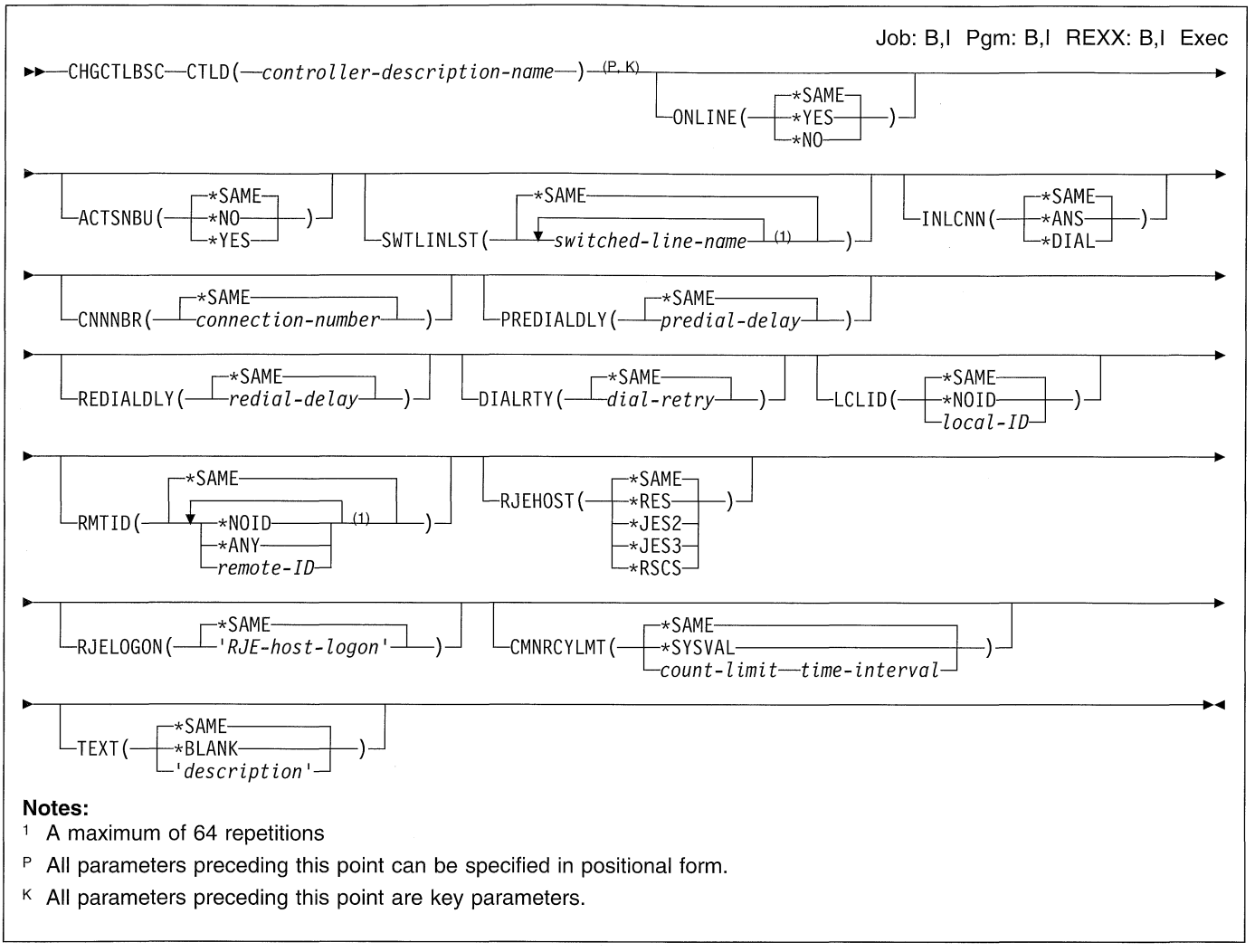
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGCTLASC  CTLD(VRTCTL325)  CNNNBR(2825555)
```

This command changes the connection number for controller description VRTCTL325 to 2825555.

CHGCTLBSC (Change Controller Description (BSC)) Command



Notes:

1 A maximum of 64 repetitions

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Controller Description (BSC) (CHGCTLBSC) command changes a controller description for a binary synchronous communications (BSC) controller.

Required Parameter

CTLD

Specifies the name of the controller description being changed.

Optional Parameters

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The controller is automatically varied on at IPL.

***NO:** This controller is not automatically varied on at IPL.

ACTSNBU

Specifies, for controllers supporting the switched network backup (SNBU) feature, whether the SNBU feature is activated or deactivated. Both the local and remote modems must support the SNBU feature to perform a valid activation.

***SAME:** The value does not change.

***NO:** The switched network backup (SNBU) feature is not activated.

***YES:** The switched network backup feature (SNBU) is activated. To start the feature when the nonswitched connection is broken, specify ACTSNBU(*YES) on the line description command.

SWTLINLST

Specifies the names of the switched lines to which this controller attaches. The line descriptions must already exist. Up to 64 switched line names can be specified.

***SAME:** The value does not change.

switched-line-name: Specify the names of up to 64 lines that are connected to this controller description. The same line name can be used more than once. For each line name specified, a line description by that name must already exist.

INLCNN
Specifies the method used to establish a connection with this controller.

***SAME:** The value does not change.

***ANS:** The connection is made by the AS/400 system when it answers an incoming call from this controller. If a call is received from the remote controller and all necessary conditions are met, the incoming call is answered by the system.

For X.25 connections, the line to which the controller attaches requires switched virtual circuits (SVC) configured on the LGLCHLE parameter of type IN or BOTH(*SVCIN or *SVCBOTH) for the connection to succeed. The line can be changed using the Change Line Description (X.25) (CHGLINX25) command.

***DIAL:** The connection is made by a call initiated from the AS/400 system.

For X.25 connection, the line attached to the controller requires switched virtual circuits (SVCs) configured on the LGLCHLE parameter of type OUT or BOTH(*SVCOUT or *SVCBOTH) for the connection to succeed.

CNNNBR
Specifies the telephone number to dial to connect to this controller.

***SAME:** The value does not change.

connection-number: Specify the telephone number used to call this controller.

PREDIALDLY
Specifies how long to wait (in 0.5 second intervals) before dialing.

***SAME:** The predial delay does not change.

predial-delay: Specify a value ranging from 1 through 254 in 0.5 second intervals, or specify 0 to indicate no delay.

REDIALDLY
Specifies how long to wait (in 0.5 second intervals) before re-dialing when the call attempt is unsuccessful.

***SAME:** The value does not change.

redial-delay: Specify a value ranging from 1 through 254 in 0.5 second intervals, or specify 0 to indicate no delay.

DIALRTY
Specifies the number of re-dial attempts made by the system before considering the dialing unsuccessful.

***SAME:** The value does not change.

dial-retry: Specify a value ranging from 0 through 254 for the number of dial attempts.

LCLID
Specifies the local ID that is used to identify the system to the remote controller.

***SAME:** The value does not change.

***NOID:** The AS/400 system sends a null identifier.

local-ID: Specify the local ID as an even number of hexadecimal characters ranging from 4 through 30 characters in length. If only four characters are specified, the first two and last two must the same, for example, F3F3 or 8484.

RMTID
Specifies a list of identifiers for remote BSC controllers. Each entry is an even number of hexadecimal characters, ranging from 4 through 30 characters in length. If only four characters are specified, the first two and last two must the same, such as F3F3 or 8484. Up to 64 remote controller IDs can be specified.

***SAME:** The value does not change.

***NOID:** The AS/400 system accepts a null identifier.

***ANY:** The AS/400 system accepts any identifier sent by the remote controller. It must be the last, or only, entry in the list.

remote-ID: Specify the identifier or a list of identifiers (64 maximum) used by remote BSC controllers. The identifier must be specified in hexadecimal form and cannot contain BSC control characters.

RJEHOST
Specifies the subsystem type of the host to which the remote job entry (RJE) function is connected.

***SAME:** The value does not change.

***RES:** The host system is a Remote Entry System (RES).

***JES2:** The host system is a Job Entry Subsystem 2 (JES2).

***JES3:** The host system is a Job Entry Subsystem 3 (JES3).

***RSCS:** The host system is a Remote Spooling Communications System (RSCS).

RJELOGON
Specifies the sign-on information for the RJE host system.

***SAME:** The value does not change.

'RJE-host-logon': Specify up to 80 characters of text enclosed in apostrophes used as sign-on (logon) information for the RJE host system.

CMNRCYLMT
Specifies the number of recovery attempts made by the system before an inquiry message is sent to the system operator. Also specifies the time (in minutes) that must

elapse before the system sends an inquiry message to the system operator indicating that the recovery attempt count limit is reached.

***SAME:** The value does not change.

***SYSVAL:** The recovery limits specified in the QCMNRCYLMT system value are used.

Element 1: Maximum Recovery Limit

count-limit: Specify the number of second-level recovery attempts to be made. Valid values range from 0 through 99.

Element 2: Recovery Time Interval

time-interval: Specify the time interval (in minutes) at which the specified number of second-level recoveries are attempted. Valid values range from 0 through 120. If the value specified for *count-limit* is not 0, the value 0 specifies infinite recovery.

count-limit: Specify the number of second-level recovery attempts to be made. Valid values range from 0 through 99.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

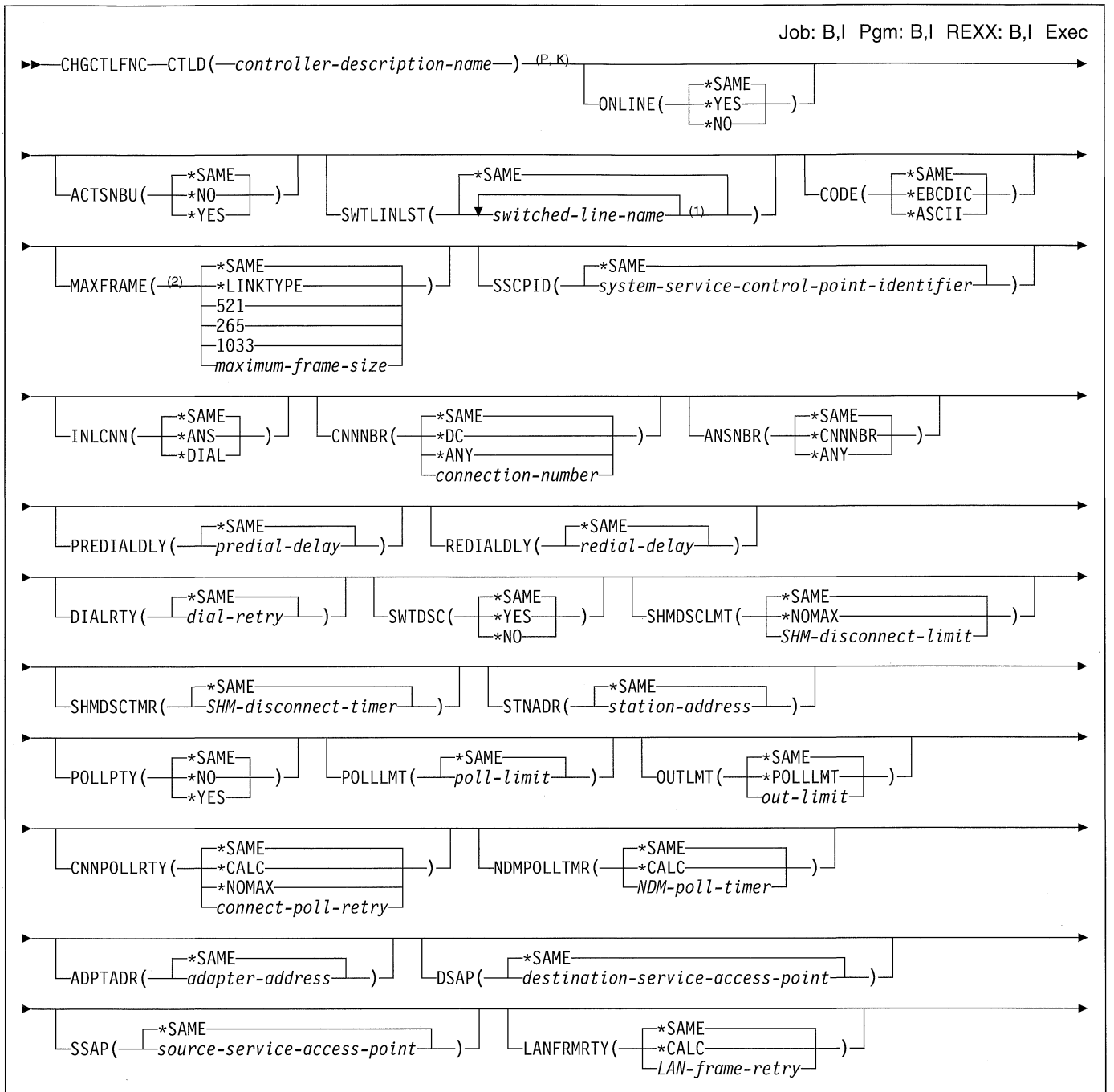
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGCTLBSC  CTLD(VRTCTL325)  LCLID(ABAB)
```

This command changes the controller named VRTCTL325 to have the local identifier ABAB.

CHGCTLFNC (Change Controller Description (Finance)) Command



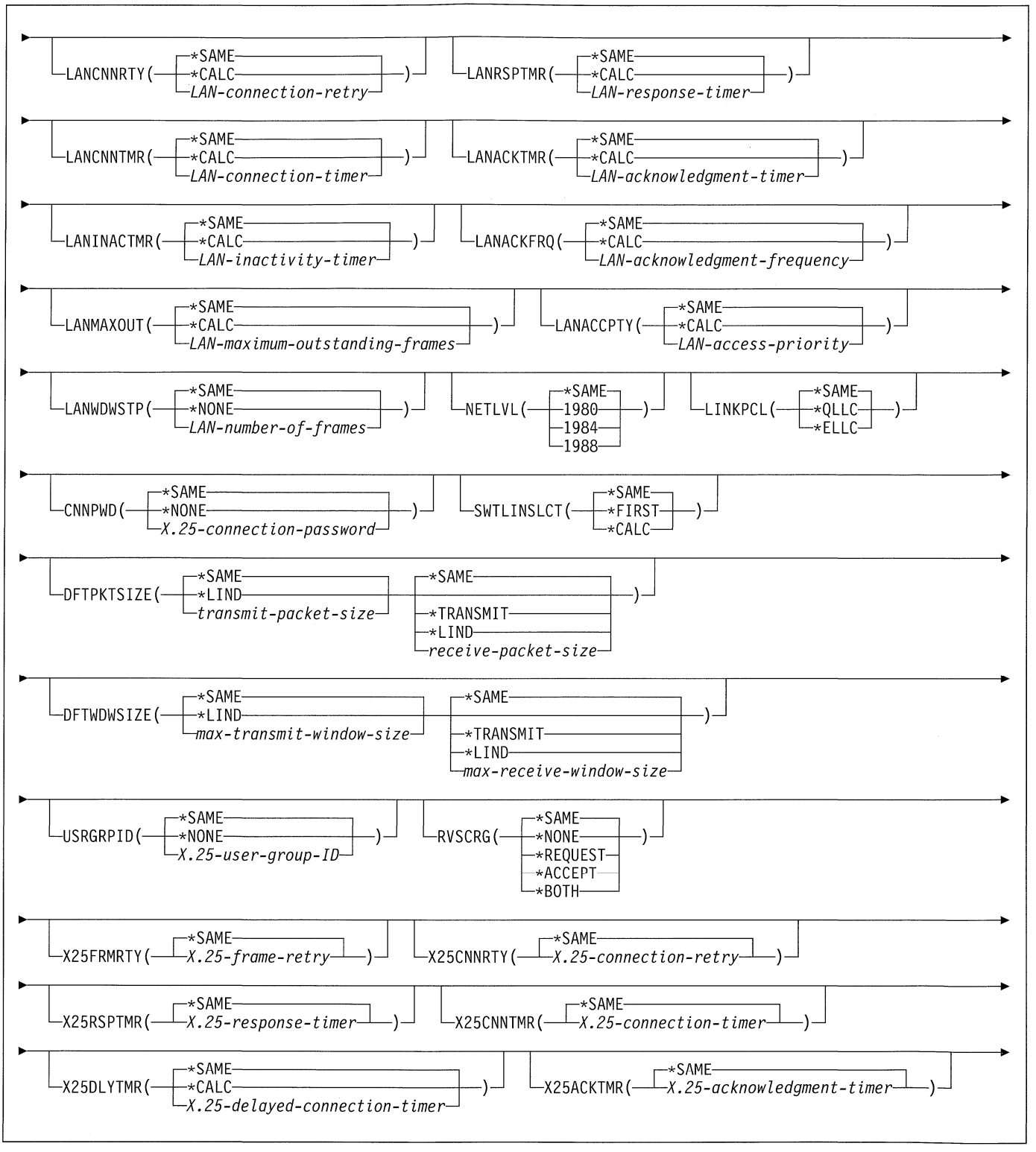
Notes:

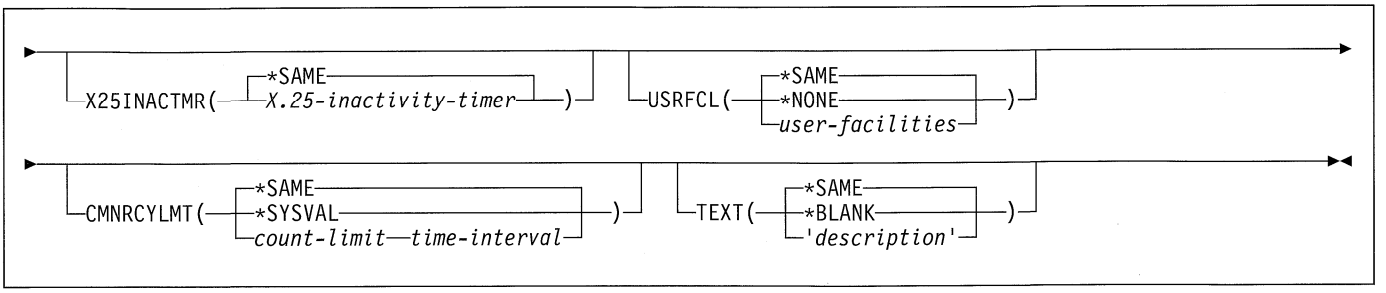
P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

1 A maximum of 64 repetitions

2 If the value of parameter LINKTYPE is *LAN, and *FBSS was specified for TYPE (on the CRTCTLFNC command), any value ranging from 265 through 521 can be specified for MAXFRAME.


LANCNNRTY(*SAME *CALC LAN-connection-retry)
LANRSPTMR(*SAME *CALC LAN-response-timer)
LANCNNTMR(*SAME *CALC LAN-connection-timer)
LANACKTMR(*SAME *CALC LAN-acknowledgment-timer)
LANINACTMR(*SAME *CALC LAN-inactivity-timer)
LANACKFRQ(*SAME *CALC LAN-acknowledgment-frequency)
LANMAXOUT(*SAME *CALC LAN-maximum-outstanding-frames)
LANACCPTY(*SAME *CALC LAN-access-priority)
LANWDWSTP(*SAME *NONE LAN-number-of-frames)
NETLVL(*SAME 1980 1984 1988)
LINKPCL(*SAME *QLLC *ELLC)
CNNPWD(*SAME *NONE X.25-connection-password)
SWTLINSLCT(*SAME *FIRST *CALC)
DFTPKTSIZE(*SAME *LIND transmit-packet-size *SAME *TRANSMIT *LIND receive-packet-size)
DFTWDWSIZE(*SAME *LIND max-transmit-window-size *SAME *TRANSMIT *LIND max-receive-window-size)
USRGRPID(*SAME *NONE X.25-user-group-ID)
RVSCRG(*SAME *NONE *REQUEST *ACCEPT *BOTH)
X25FRMRTY(*SAME X.25-frame-retry)
X25CNNRTY(*SAME X.25-connection-retry)
X25RSPTMR(*SAME X.25-response-timer)
X25CNNTMR(*SAME X.25-connection-timer)
X25DLYTMR(*SAME *CALC X.25-delayed-connection-timer)
X25ACKTMR(*SAME X.25-acknowledgment-timer)



Purpose

The Change Controller Description (Finance) (CHGCTLFNC) command changes a controller description for a finance controller.

Required Parameter

CTLD
Specifies the name of the controller description being changed.

Optional Parameters

ONLINE
Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** This value does not change.

***YES:** The controller is automatically varied on at IPL.

***NO:** This controller is not automatically varied on at IPL.

ACTSNBU
Specifies, for controllers supporting the switched network backup (SNBU) feature, whether the SNBU feature is activated or deactivated. Both the local and remote modems must support the SNBU feature to perform a valid activation.

***SAME:** The value does not change.

***NO:** The switched network backup (SNBU) feature is not activated.

***YES:** The switched network backup feature (SNBU) is activated. To start the feature when the nonswitched connection is broken, specify ACTSNBU(*YES) on the line description command.

SWTLINLST
Specifies the names of the switched lines to which this controller attaches. The line descriptions must already exist. Up to 64 switched line names can be specified.

***SAME:** The value does not change.

switched-line-name: Specify the names of up to 64 lines that are connected to this controller description. The same line name can be used more than once. For each line name specified, a line description with that name must already exist.

CODE
Specifies the character code used. The code can be either extended binary-coded decimal interchange code (*EBCDIC) or the American National Standard Code for Information Interchange (*ASCII).

***SAME:** The value does not change.

***EBCDIC:** The extended binary-coded decimal interchange code (EBCDIC) character set code is used.

***ASCII:** The ASCII character set code is used.

MAXFRAME
Specifies the maximum frame (path information unit (PIU)) size that the controller can send or receive. This value is used to calculate request unit (RU) sizes. Since the maximum PIU size that the controller can send or receive is negotiated at exchange identifier time, the maximum PIU size used at run time may be different. This value matches the corresponding value on the host system.

***SAME:** The maximum frame size does not change.

***LINKTYPE:** The following frame sizes (measured in bytes) are specified if *LINKTYPE is used:

*SDLC - 265 *LAN - 521 *X25 - 256

maximum-frame-size: Specify a maximum frame size for the controller. The frame size that can be used depends on the controller and type of line being used. Valid frame sizes are shown below.

Table 9. Maximum Frame Sizes (in bytes)

Controllers	Link Type	Frame Sizes
*FBSS	*LAN	265 - 521
4701, 4702, *FBSS	*SDLC	265, 521, 1033
4701, 4702, *FBSS	*X25	265, 521, 256, 512
3694, 4730, 4731, 4732, 4746	*SDLC	265

SSCPID
Specifies the system service control point (SSCP) of the host system.

***SAME:** The value does not change.

system-service-control-point-identifier: Specify the SSCP identifier as a 12-digit hexadecimal value.

INLCNN

Specifies the method used to make the initial connection on a switched line between the system and the remote controller. For X.25 lines, this parameter specifies the type of switched virtual circuit (SVC) connection that is made over the X.25 line.

***SAME:** The value does not change.

***ANS:** The connection is made by the AS/400 system when it answers an incoming call from this controller. If a call is received from the remote controller and all necessary conditions are met, the incoming call is answered by the system.

For X.25 connections, the line to which the controller attaches requires switched virtual circuits (SVC) configured on the LGLCHLE parameter of type IN or BOTH(*SVCIN or *SVCBOTH) for the connection to succeed. The line can be changed using the Change Line Description (X.25) (CHGLINX25) command.

***DIAL:** The connection is made by a call initiated from the AS/400 system.

For X.25 connection, the line attached to the controller requires switched virtual circuits (SVCs) configured on the LGLCHLE parameter of type OUT or BOTH(*SVCOUT or *SVCBOTH) for the connection to succeed.

CNNNBR

Specifies the telephone number to dial to connect to this controller.

***SAME:** The value does not change.

***DC:** For X.21 circuit switched connections, a direct call is used to connect to the controller.

***ANY:** Calls are accepted from any network address.

connection-number: Specify the connection number used to call this controller. This could be a telephone number, an X.25 network address, or an X.21 connection number depending on the type of controller and the type of line to which it is attached.

ANSNBR

Specifies the X.25 network address from which to accept calls.

***SAME:** The value does not change.

***CNNNBR:** Calls from the X.25 network address specified by the connection number (CNNNBR) parameter are accepted.

***ANY:** Calls are accepted from any X.25 network address.

PREDIALDLY

Specifies how long to wait (in 0.5 second intervals) before dialing.

***SAME:** The amount of time does not change.

predial-delay: Specify a value ranging from 1 through 254 in 0.5-second intervals, or specify 0 to indicate no delay.

REDIALDLY

Specifies how long to wait (in 0.5 second intervals) before re-dialing when the call attempt is unsuccessful.

***SAME:** The value does not change.

redial-delay: Specify a value ranging from 1 through 254 in 0.5-second intervals, or specify 0 to indicate no delay.

DIALRTY

Specifies the number of re-dial attempts made by the system before considering the dialing unsuccessful.

***SAME:** The value does not change.

dial-retry: Specify a value ranging from 0 through 254 for the number of dial attempts.

SWTDSC

Specifies whether the switched connection to this controller is made inoperative when the last device is varied off.

***SAME:** The value does not change.

***YES:** The switched connection is disconnected when the last device is varied off.

***NO:** The switched connection is not made inoperative when the last device is varied off.

SHMDSCLMT

Specifies the number of nonproductive responses (RR or RNR) that are required from the remote station before the connection can be suspended for this X.21 short-hold mode connection. This parameter is used only if SHM(*YES) is specified.

***SAME:** The value does not change.

***NOMAX:** There is no disconnect limit.

SHM-disconnect-limit: Specify a number ranging from 1 through 254 that indicates the number of nonproductive responses that must be received before the connection can be suspended.

SHMDSCTMR

Specifies the minimum length of time that the primary system maintains the connection to the remote system for this X.21 short-hold mode controller, in tenths of a second. This parameter is used only if SHM(*YES) is specified.

***SAME:** The value does not change.

SHM-disconnect-timer: Specify a value ranging from 2 through 3000 that indicates the minimum length of time, in tenths of a second, that the primary system maintains the connection to the remote system.

STNADR

Specifies the station address used when communicating with the controller.

***SAME:** The value does not change.

station-address: Specify a hexadecimal value ranging from 01 through FE. This is the station address of the remote controller.

POLLPTY

Specifies whether this controller has priority when polled.

***SAME:** The value does not change.

***NO:** This controller does not have polling priority.

***YES:** This controller has polling priority.

POLLLMT

Specifies the number of consecutive polls that are issued to the same controller when the poll results in receiving frames.

***SAME:** The value does not change.

poll-limit: Specify a value ranging from 0 through 4 (0 meaning no consecutive polls are attempted) for the number of consecutive polls.

OUTLMT

Specifies the number of times SDLC allows the consecutive transmission of the maximum number of frames to a station before allowing transmission to another station.

***SAME:** The value does not change.

***POLLLMT:** The value specified on the POLLLMT parameter is used.

out-limit: Specify a value ranging from 0 through 4 for the number of consecutive transmissions.

CNNPOLLRTY

Specifies the number of connect poll retries.

***SAME:** The value does not change.

***CALC:** The number of retries is 7 if the controller is switched, and *NOMAX if the controller is nonswitched.

***NOMAX:** There is no disconnect limit.

connect-poll-retry: Specify a value ranging from 0 through 65534 for the number of retries.

NDMPOLLTMR

Specifies the slow poll interval for this controller when it is in normal disconnect mode (NDM).

***SAME:** The value does not change.

***CALC:** The system determines the value to use.

NDM-poll-timer: Specify a value ranging from 1 through 3000 in 0.1-second intervals, or specify 0 to indicate no timer.

ADPTADR

Specifies the 12-character adapter address of the remote controller. This is the address to which the system sends data when it communicates with the remote controller. This value can be obtained from the remote controller's configuration record. Valid values range from hex 000000000001 through hex FFFFFFFFFFFF.

***SAME:** The value does not change.

adapter-address: Specify the adapter address.

DSAP

Specifies the logical address of the SNA destination service access point (DSAP) to which this system sends data when it communicates with the remote controller. This address allows the controller and the system to route the data that comes from this system. The value must match the value specified on the source service access point (SSAP) parameter in the remote controller's configuration record.

***SAME:** The value does not change.

destination-service-access-point: Specify a hexadecimal value ranging from 04 through 9C, in increments of 4 (for example, 04, 08, 0C, 10) to represent the destination service access point.

SSAP

Specifies the source service access point. This is the logical address this system uses when it sends data to the remote controller. This address allows the remote controller and the system to route the data that comes from this system. It must match the value assigned to the destination service access point (DSAP) prompt in the remote controller's configuration record.

***SAME:** The value does not change.

source-service-access-point: Specify a hexadecimal value ranging from 04 through 9C, in increments of 4 (for example, 04, 08, 0C, 10) to represent the source service access point.

LANFRMRTY

Specifies the number of times a frame is transmitted if there is no acknowledgment from the remote controller in the time period specified by the LANRSPTMR parameter. This value is only used after a successful connection has been made.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-frame-retry: Specify a value ranging from 0 through 254 for the number of times a frame is transmitted before an acknowledgement is received.

LANCNNRTY

Specifies the number of times a transmission is attempted before an acknowledgement is received. This value is used at connection time (unlike LANFRMRTY which is used after a connection has been made).

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-connection-retry: Specify a value ranging from 0 through 254 for the number of times the transmission is attempted before an acknowledgement is received.

LANRSPTMR
Specifies the length of time the system waits before an inoperative condition occurs on a link after a connection is made.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-response-timer: Specify a value ranging from 1 through 254 in 0.1-second intervals for the amount of time the system waits before an inoperative condition occurs.

LANCNNTMR
Specifies the length of time the system waits before an inoperative condition occurs on a link at connection time.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-connection-timer: Specify a value ranging from 1 through 254 in 0.1-second intervals for the amount of time the system waits before an inoperative condition occurs. The value 0 indicates that a 6-second connection timer is used.

LANACKTMR
Specifies the length of time the system waits before sending an acknowledgement for received frames.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-acknowledgment-timer: Specify a value ranging from 1 through 254 in 0.1-second intervals, or specify 0 to indicate no delay. If 0 is specified for this parameter, 0 must also be specified for the LANACKFRQ parameter. If a non zero value is specified for this parameter, a non zero value must also be specified for the LANACKFRQ parameter.

LANINACTMR
Specifies the length of time used to determine an inactive condition for the controller.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-inactivity-timer: Specify a value ranging from 1 through 255 in 0.1-second intervals for the length of time used to determine an inactive condition for the controller. The value 0 indicates that a 6-second connection time is used.

LANACKFRQ
Specifies the maximum number of frames that are received before an acknowledgment is sent to the controller.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-acknowledgment-frequency: Specify a value ranging from 0 through 127 for the number of frames received. If 0 is specified for this parameter, 0 must also be specified for the LANACKTMR parameter. If a non zero value is specified for this parameter, a non zero value must also be specified for the LANACKTMR parameter.

LANMAXOUT
Specifies the maximum number of frames that can be sent before an acknowledgment is received from the remote system.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-maximum-outstanding-frames: Specify a value ranging from 1 through 127 for the number of frames that can be sent before an acknowledgement is received.

LANACCPTY
Specifies the priority granted to the sending system for sending frames. The larger the number, the higher the priority.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-access-priority: Specify a value ranging from 0 through 3 to indicate how soon a frame can be sent on a local area network.

LANWDWSTP
Specifies whether to reduce the number of outstanding frames that may be sent before an acknowledgement is received from the remote system during network congestion. This parameter (LAN Window Step) also indicates the number of frames that must be successfully received before the number of outstanding frames can be incremented to the maximum allowable value.

***SAME:** The value does not change.

***NONE:** The number of outstanding frames is not reduced during network congestion.

LAN-number-of-frames: Specify the number of outstanding frames to be reduced before an acknowledgement is received from the remote system.

NETLVL
Specifies the level of the X.25 network used to reach this controller. The level is specified by giving the year of the standard of the X.25 network.

***SAME:** The value does not change.

1980: The 1980 standard is used.

1984: The 1984 standard is used.

1988: The 1988 standard is used.

LINKPCL

Specifies the link level protocol used on the X.25 network to communicate with this controller.

***SAME:** The value does not change.

***QLLC:** The QLLC protocol is used.

***ELLC:** The ELLC protocol is used.

CNNPWD

Specifies, for X.25 switched virtual circuit (SVC) controllers, the password used when connecting to this controller.

***SAME:** The value does not change.

***NONE:** No password is used.

X.25-connection-password: Specify the connection password. The password for each controller can consist of any alphanumeric characters represented by the hexadecimal values ranging from 40 through FF.

SWTLINSLCT

Specifies the method that is used to select lines from an X.25 switched line list.

***SAME:** The value does not change.

***FIRST:** The lines are selected in the order in which they are specified.

***CALC:** The system determines the value to use.

DFTPKTSIZE

Specifies the default packet size used by the X.25 network.

Element 1: Transmit Packet Size

***SAME:** The value does not change.

***LIND:** The value specified in the line description is used as the default packet size.

transmit-packet-size: Specify a default packet size for transmission. The valid values for the packet size are 64, 128, 256, 512, 1024, 2048, and 4096.

Element 2: Receive Packet Size

***SAME:** The value does not change.

***LIND:** The value specified in the line description is used as the default packet size.

***TRANSMIT:** The value specified as the default packet size for transmission is used as the default for reception.

receive-packet-size: Specify a default packet size for reception. The valid values for the packet size are 64, 128, 256, 512, 1024, 2048, and 4096.

DFTWDWSIZE

Specifies the default window size used by the X.25 network.

Element 1: Transmit Window Size

***SAME:** The value does not change.

***LIND:** The value specified in the line description is used as the default window size.

max-transmit-window-size: Specify the appropriate default window size. Valid values range from 1 through 7 for networks that use modulus 8 packet numbering. Valid values range from 1 through 15 for networks that use modulus 128 packet numbering.

Element 2: Receive Window Size

***SAME:** The value does not change.

***LIND:** The value specified in the line description is used as the default window size.

***TRANSMIT:** The value specified as the default window size for transmission is used as the default for reception.

max-receive-window-size: Specify the appropriate default window size. Valid values range from 1 through 7 for networks that use modulus 8 packet numbering. Valid values range from 1 through 15 for networks that use modulus 128 packet numbering.

USRGRPID

Specifies the closed user-group ID for contacting this X.25 switched virtual circuit (SVC) controller. Specify a value ranging from 0 through 99, as provided by the network subscription. This parameter is not valid for permanent virtual circuit (PVC) connections. It is valid only for SVC circuit outgoing call operations and is ignored for SVC incoming call connections.

***SAME:** The value does not change.

***NONE:** No user group identifier is specified.

X.25-user-group-ID: Specify the closed user-group ID for contacting an X.25 SVC controller. Valid values range from 0 through 99, as provided by the network subscription.

RVSCRG

Specifies whether reverse charges are accepted or requested when contacting this controller.

***SAME:** The value does not change.

***NONE:** No reverse charging for network tariff billing is accepted.

***REQUEST:** Charges are requested on outgoing call request packets.

***ACCEPT:** Reverse charging for network tariff billing is accepted on incoming requests.

***BOTH:** Both incoming and outgoing requests are accepted.

X25FRMRTY

Specifies the maximum number of times a frame is sent after the response timer ends when connected to this controller.

***SAME:** The value does not change.

X.25-frame-retry: Specify a value ranging from 0 through 21 for the number of times a frame is sent.

X25CNNRTY
Specifies the maximum number of times that a frame is sent after the connect response timer ends when connecting to this controller.

***SAME:** The value does not change.

X.25-connection-retry: Specify a value ranging from 0 through 21 for the retry value.

X25RSPTMR
Specifies the time allowed to return an acknowledgment when a frame is sent while connected to this controller.

***SAME:** The value does not change.

X.25-response-timer: Specify a value ranging from 1 through 2550 in 0.1-second intervals.

X25CNNTMR
Specifies the time allowed to return an acknowledgment when a frame is sent while connecting to this controller.

***SAME:** The value does not change.

X.25-connection-timer: Specify a value ranging from 1 through 2550 in 0.1-second intervals.

X25DLYTMR
Specifies the time interval to try establishing a connection to the controller.

***SAME:** The value does not change.

***CALC:** The values specified for the X25CNNTMR and X25CNNRTY parameters are used to determine how often and how many times to try establishing the connection.

X.25-delayed-connection-timer: Specify a value ranging from 1 through 32767 in 0.1-second intervals. The system retries the connection indefinitely at the specified interval.

X25ACKTMR
Specifies the time period to delay sending acknowledgments for received frames.

***SAME:** The value does not change.

X.25-acknowledgment-timer: Specify a value ranging from 1 through 2550 in 0.1-second intervals, or specify 0 to indicate no delay.

X25INACTMR
Specifies the time period used to determine an inactive condition for the controller.

***SAME:** The value does not change.

X.25-inactivity-timer: Specify a value ranging from 1 through 2550 in 0.1-second intervals.

USRFCL
Specifies a string of hexadecimal digits sent to the X.25 network to request additional services. The AS/400 system allows up to 218 hexadecimal characters.

***SAME:** The value does not change.

***NONE:** No additional services are needed.

user-facilities: Specify a string of hexadecimal digits sent to the network to request additional services. The AS/400 system allows up to 218 hexadecimal characters.

CMNRCYLMT
Specifies the number of recovery attempts made by the system before an inquiry message is sent to the system operator. Also specifies the time (in minutes) that must elapse before the system sends an inquiry message to the system operator indicating that the recovery attempt count limit is reached.

***SAME:** The value does not change.

***SYSVAL:** The recovery limits specified in the QCMNRCYLMT system value are used.

Element 1: Maximum Recovery Limit

count-limit: Specify the number of second-level recovery attempts to be made. Valid values range from 0 through 99.

Element 2: Recovery Time Interval

time-interval: Specify the time interval (in minutes) at which the specified number of second-level recoveries are attempted. Valid values range from 0 through 120. If the value specified for *count-limit* is not 0, the value 0 specifies infinite recovery.

TEXT
Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

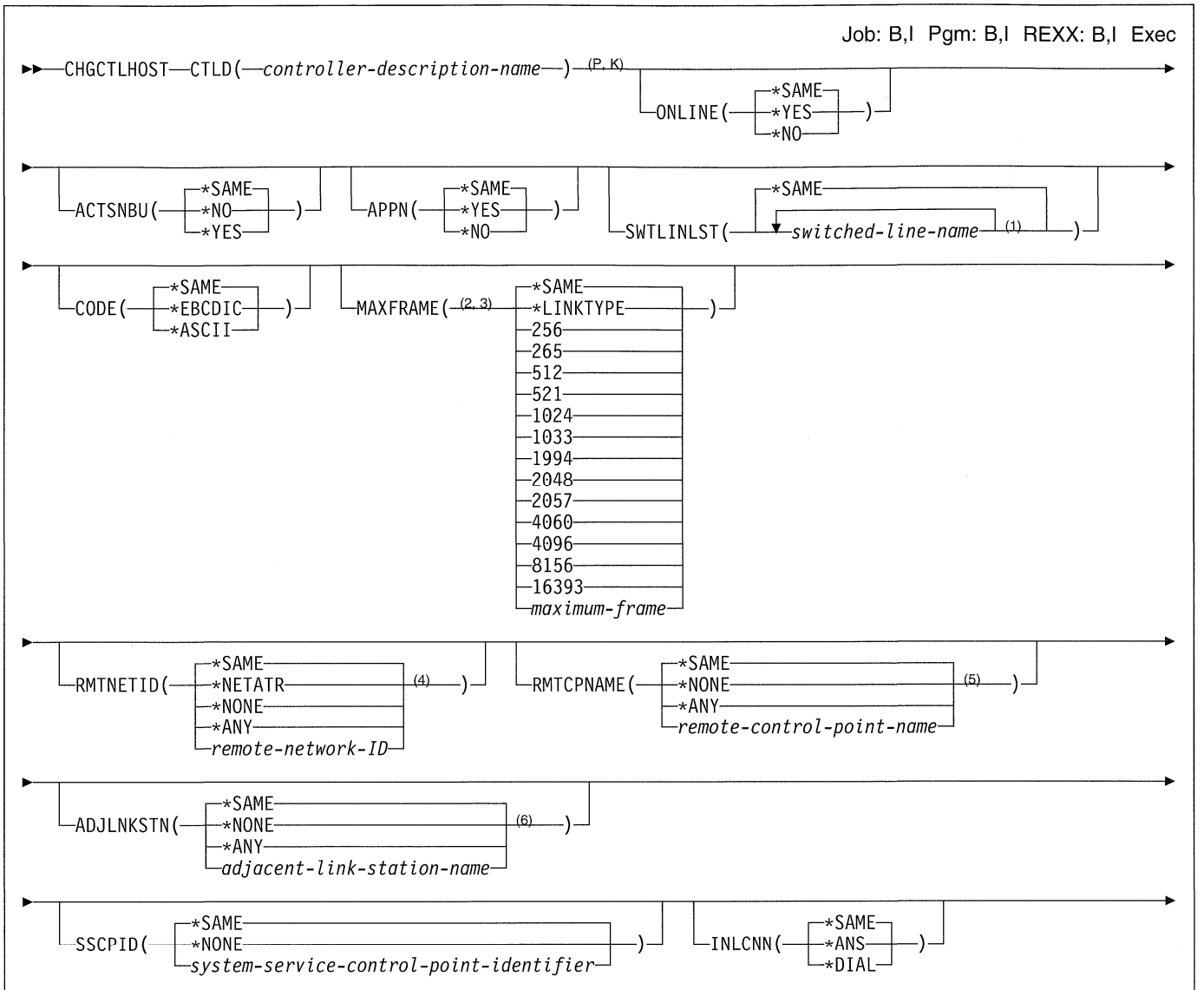
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGCTLFNC  CTLD(FNC1)  X25FRMRTY(7)
```

This command changes the controller named FNC1 to have an X.25 frame retry of seven.

CHGCTLHOST (Change Controller Description (SNA Host)) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

1 A maximum of 64 repetitions

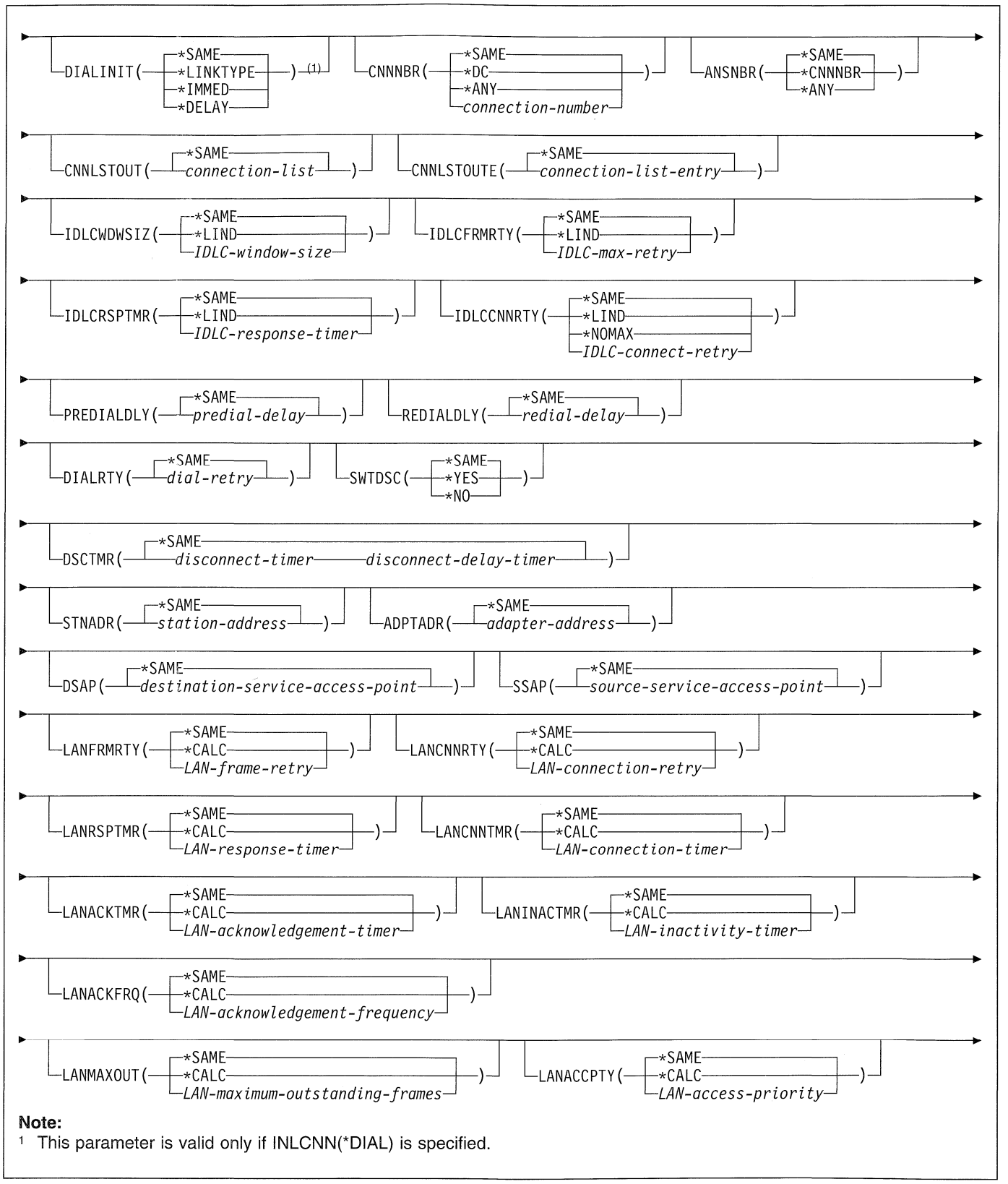
2 If LINKTYPE is *LAN, valid values range from 265 through 16393.

3 For DDI LANs, valid values range from 265 through 4444.

4 If RMTNETID(*ANY) is specified, RMTCPNAME(*ANY) must also be specified.

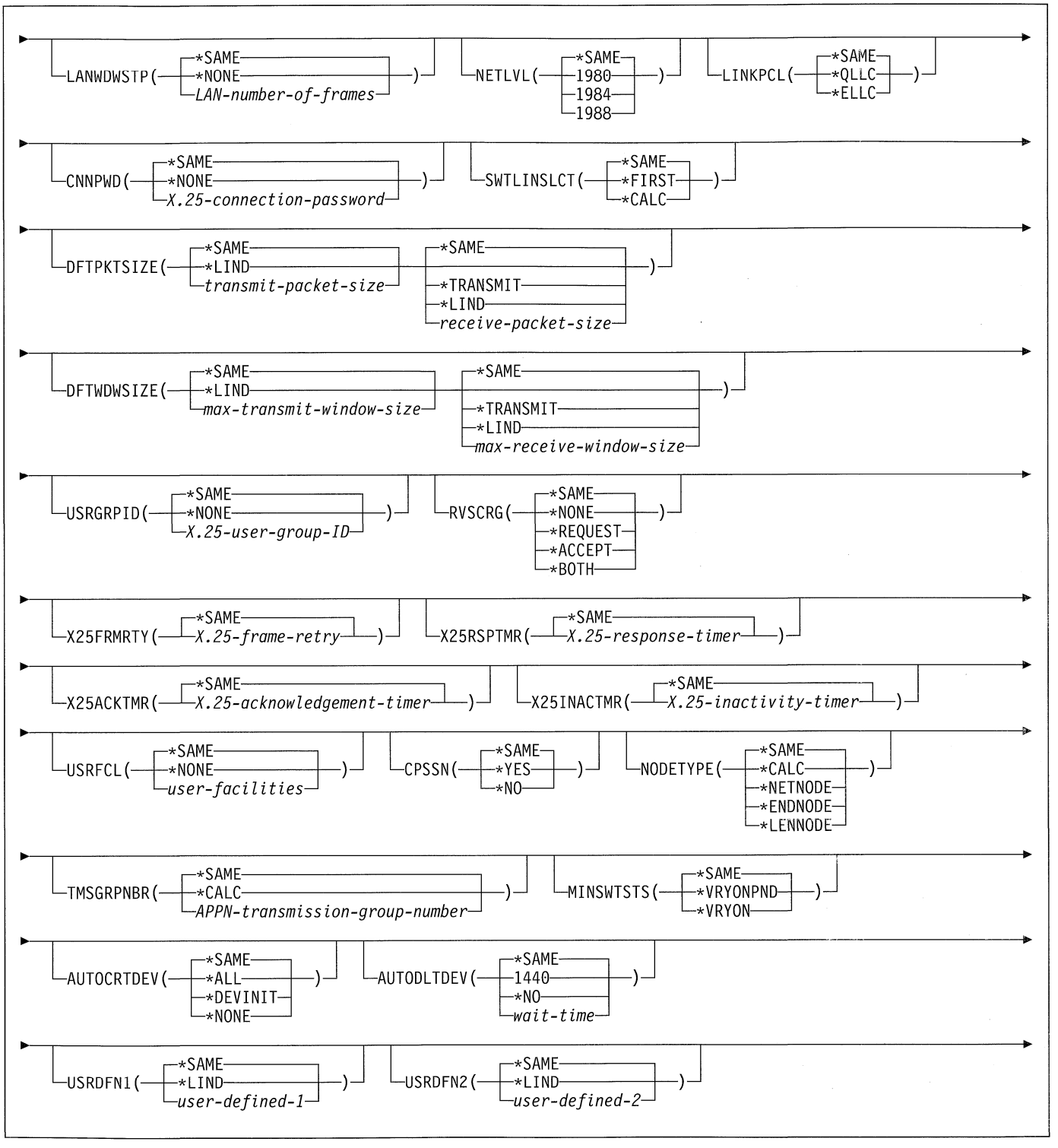
5 RMTCPNAME(*ANY) cannot be specified if LINKTYPE is *IDLC and SWITCHED or SNBU is *YES; if LINKTYPE is *SDLC and SWITCHED or SNBU is *YES; if NODETYPE is *LENNODE; or if MDLCTL is *YES.

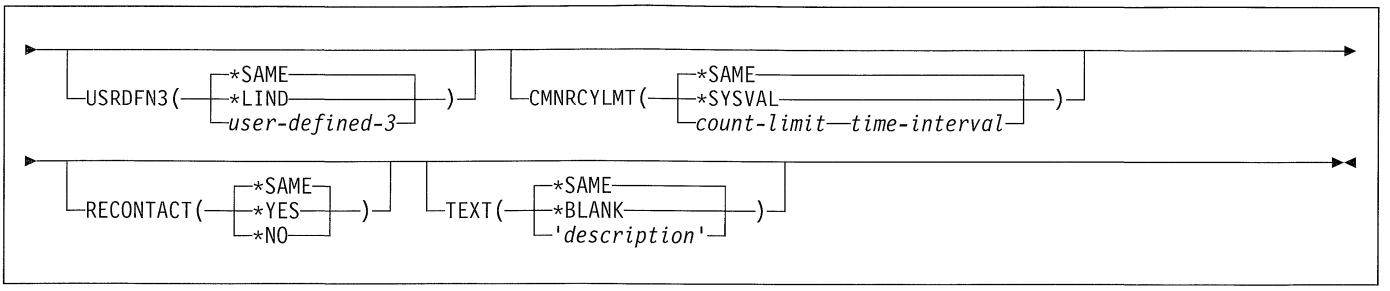
6 ADJLNKSTN(*ANY) is only valid if SWITCHED is *YES and LINKTYPE is *SDLC or *IDLC. Also, ADJLNKSTN must be specified if RMTCPNAME is *ANY.



Note:

1 This parameter is valid only if INLCNN(*DIAL) is specified.


LANWDWSTP(*SAME *NONE LAN-number-of-frames)
NETLVL(*SAME 1980 1984 1988)
LINKPCL(*SAME *QLLC *ELLC)
CNNPWD(*SAME *NONE X.25-connection-password)
SWTLINSLCT(*SAME *FIRST *CALC)
DFTPKTSIZE(*SAME *LIND transmit-packet-size *SAME *TRANSMIT *LIND receive-packet-size)
DFTWDWSIZE(*SAME *LIND max-transmit-window-size *SAME *TRANSMIT *LIND max-receive-window-size)
USRGRPID(*SAME *NONE X.25-user-group-ID)
RVSCRG(*SAME *NONE *REQUEST *ACCEPT *BOTH)
X25FRMRTY(*SAME X.25-frame-retry)
X25RSPTMR(*SAME X.25-response-timer)
X25ACKTMR(*SAME X.25-acknowledgement-timer)
X25INACTMR(*SAME X.25-inactivity-timer)
USRFCL(*SAME *NONE user-facilities)
CPSSN(*SAME *YES *NO)
NODETYPE(*SAME *CALC *NETNODE *ENDNODE *LENNODE)
TMSGRPNBR(*SAME *CALC APPN-transmission-group-number)
MINSWTSTS(*SAME *VRYONPND *VRYON)
AUTOCRTDEV(*SAME *ALL *DEVINIT *NONE)
AUTODLTDEV(*SAME 1440 *NO wait-time)
USRDFN1(*SAME *LIND user-defined-1)
USRDFN2(*SAME *LIND user-defined-2)



Purpose

The Change Controller Description (SNA Host) (CHGCTLHOST) command changes a controller description for a Systems Network Architecture (SNA) host system.

Required Parameter

CTLD
Specifies the name of the controller description being changed.

Optional Parameters

ONLINE
Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The controller is automatically varied on at IPL.

***NO:** This controller is not automatically varied on at IPL.

ACTSNBU
Specifies, for controllers supporting the switched network backup (SNBU) feature, whether the SNBU feature is activated or deactivated. Both the local and remote modems must support the SNBU feature to perform a valid activation.

***SAME:** The value does not change.

***NO:** The switched network backup (SNBU) feature is not activated.

***YES:** The switched network backup feature (SNBU) is activated. To start the feature when the nonswitched connection is broken, specify ACTSNBU(*YES) on the line description command.

APPN
Specifies whether the local system uses advanced peer-to-peer networking (APPN) functions when communicating with this controller. If APPN functions are used, you must specify a remote control point name and a remote network identifier.

***SAME:** The value does not change.

***YES:** The local system uses APPN functions.

***NO:** The local system does not use APPN functions.

SWTLINLST
Specifies the names of the switched lines to which this controller attaches. The line descriptions must already exist. Up to 64 switched line names can be specified.

Note: The same line name can be used more than once.

***SAME:** The value does not change.

switched-line-name: Specify the name of a line for which a line description already exists. The maximum number of lines that can be connected to this controller is 64.

CODE
Specifies the character code used. The code can be either extended binary-coded decimal interchange code (*EBCDIC) or the American National Standard Code for Information Interchange (*ASCII).

***SAME:** The value does not change.

***EBCDIC:** The extended binary-coded decimal interchange code (EBCDIC) character set code is used.

***ASCII:** The ASCII character set code is used.

MAXFRAME
Specifies the maximum frame (path information unit (PIU)) size that the controller can send or receive. This value is used to calculate request unit (RU) sizes. Since the maximum PIU size that the controller can send or receive is negotiated at exchange identifier time, the maximum PIU size used at run time may be different. This value matches the corresponding value on the host system.

***SAME:** The value does not change.

***LINKTYPE:** The following values are used when *LINKTYPE is specified:

```
*SDLC-521    *LAN-16393    *FR-1590
*IDLC-2048   *X25-1024
```

maximum-frame: Specify the maximum frame size for the controller. The frame size that can be used depends on the type of line being used. Valid frame sizes for each line type are shown below.

Link Type	Frame Size (in bytes)
***FR**	265 - 8182
***LAN**	265 - 16393 (265 - 4444 for DDI LANs)
***IDLC**	265 - 8196

*SDLC	265,521,1033,2057
*X25	256,265,512,521,1024,1033,2048, 4096

RMTNETID
Specifies the identifier (ID) of the remote network.

***SAME:** The value does not change.

***NETATR:** The RMTNETID value specified in the system network attributes is used.

***NONE:** No remote network identifier (ID) is used.

***ANY:** The system determines which remote network identifier is used.

remote-network-ID: Specify the remote network ID.

RMTCPNAME
Specifies the name of the remote control point.

***SAME:** The value does not change.

***NONE:** No remote control point name is used.

***ANY:** The system determines the name of the remote control point to use.

remote-control-point-name: Specify the remote control point name.

ADJLNKSTN
Specifies the name of the adjacent link station. This name is used by the system to identify which switched controller description on the AS/400 system is used to establish a link to a host S/370 system. The adjacent link station name for the AS/400 system must match the name provided by the host system during link activation.

***SAME:** The value does not change.

***NONE:** No adjacent link station name is specified.

***ANY:** The system determines which adjacent link station is used.

adjacent-link-station-name: Specify the adjacent link station name.

SSCPID
Specifies the system service control point (SSCP) of the host system.

***SAME:** The value does not change.

***NONE:** No SSCP identifier is specified.

system-service-control-point-identifier: The SSCP identifier is a 12-digit hexadecimal value, with the first 2 digits being hexadecimal 05.

INLCNN
Specifies the method used to establish a connection with this controller.

***SAME:** The value does not change.

***ANS:** The connection is made by the local system answering an incoming call from this controller.

***DIAL:** The connection is made by a call initiated from the AS/400 system.

For X.25 connection, the line attached to the controller requires switched virtual circuits (SVCs) configured on the LGLCHLE parameter of type OUT or BOTH(*SVCOUT or *SVCBOTH) for the connection to succeed.

DIALINIT
Specifies the method used to make the initial dial on a switched line between the system and the remote controller.

***SAME:** The value does not change.

***LINKTYPE:** The type of dial connection initiated is specified on the LINKTYPE parameter. For LAN or SDLC short-hold mode connections, the default is to dial the connection immediately upon vary on of the controller description. For all other link types, the default is to defer the dial.

***IMMED:** The dial connection is initiated immediately upon vary on of the controller description.

***DELAY:** The dial connection is delayed until a job is initiated that requests the use of the remote controller resources.

CNNNBR
Specifies the telephone number to dial to connect to this controller.

***SAME:** The value does not change.

***DC:** For X.21 circuit switched connections, a direct call is used to connect to the controller.

***ANY:** Calls are accepted from any network address.

connection-number: Specify the connection number used to call this controller.

This could be a telephone number, an X.25 network address, or an X.21 connection number, depending on the type of controller and the type of line to which it is attached.

ANSNBR
Specifies the X.25 network address from which to accept calls.

***SAME:** The value does not change.

***CNNNBR:** Calls from the X.25 network address specified by the connection number (CNNNBR) parameter are accepted.

***ANY:** Calls are accepted from any X.25 network address.

CNNLSTOUT
Specifies, for ISDN switched connections, the name of a connection list object that contains the Public Switched Network assigned numbers for a dial out operation to the Public Switched Data Network.

***SAME:** The value does not change.

connection-list: Specify the name of a connection list object.

CNNLSTOUTE
Specifies, for IDLC switched connections, the entry name from the connection list that is used to make a call to the Public Switched Data Network. The connection list must be specified on the CNNLSTOUT parameter.

***SAME:** The value does not change.

connection-list-entry: Specify an entry name.

IDLCWDWSIZ
Specifies the window size used by the line description.

***SAME:** The value does not change.

***LIND:** The window size specified in the line description is used.

IDLC-window-size: Specify the window size. Valid values range from 1 through 31.

IDLCFRMRTY
Specifies the maximum number of attempts to transmit a frame before an error is reported.

***SAME:** The value does not change.

***LIND:** The number of attempts specified in the line description is used.

IDLC-frame-retry: Specify the number of attempts. Valid values range from 0 through 100.

IDLCRSPTMR
Specifies the amount of time, in tenths of a second, to wait before retransmitting a frame if acknowledgement has not been received.

***SAME:** The value does not change.

***LIND:** The time specified in the line description is used.

IDLC-response-timer: Specify the amount of time to wait. Valid values range from 10 through 100 tenths of a second. For example, 10 seconds equals 100 tenths of a second.

IDCCNNRTY
Specifies the number of times to attempt retransmission at connection time.

***SAME:** The value does not change.

***LIND:** The number of attempts specified in the line description is used.

***NOMAX:** There is no disconnect limit.

IDLC-connect-retry: Specify the number of attempts. Valid values range from 1 through 100.

PREDIALDLY
Specifies how long to wait (in 0.5 second intervals) before dialing.

***SAME:** The value does not change.

predial-delay: Specify a value ranging from 1 through 254 in 0.5 second intervals, or specify 0 to indicate no delay.

REDIALDLY
Specifies how long to wait (in 0.5 second intervals) before re-dialing when the call attempt is unsuccessful.

***SAME:** The value does not change.

redial-delay: Specify a value ranging from 1 through 254 in 0.5 second intervals, or specify 0 to indicate no delay.

DIALRTY
Specifies the number of re-dial attempts made by the system before considering the dialing unsuccessful.

***SAME:** The value does not change.

dial-retry: Specify a value ranging from 0 through 254 for the number of redial attempts.

SWTDSC
Specifies whether the switched connection to this controller is disconnected when the last device in the host controller description becomes inactive.

***SAME:** The value does not change.

***YES:** The switched connection is disconnected when the last device is varied off.

***NO:** This connection remains active even if the above conditions were met, or if the host system is configured to disconnect the switched connection.

Switched disconnect is valid only if *YES is specified for the SWITCHED parameter, for the SNBU parameter, or if the LINKTYPE is *LAN.

Note: The DISCNT parameter of the GROUP macro instruction in the Network Control Program/Virtual Telecommunications Access Method (NCP/VTAM) definition may be used if the host system is to determine the disconnect characteristics.

DSCTMR
Specifies options for controlling the time (in seconds) before an inactive connection is dropped (Element 1), or the amount of time to delay the automatic disconnection (Element 2).

Note: The DISCNT parameter of the GROUP macro instruction in the NCP/VTAM definition may be used if the host system is to determine the disconnect characteristics.

Element 1: Minimum Connect Timer

***SAME:** The value does not change.

disconnect-timer: Specify a time to wait before disconnecting. Valid values range from 0 through 65535 seconds.

Element 2: Disconnect Delay Timer

***SAME:** The value does not change.

disconnect-delay-timer: Specify a value to delay link take down after the last session on the controller is terminated. Valid values range from 0 through 65535 seconds.

STNADR
Specifies the station address used when communicating with the controller.

Note: This is the station address of the local system.

***SAME:** The value does not change.

station-address: Specify the station address as a hexadecimal value ranging from 01 through FE.

ADPTADR
Specifies the 12-character adapter address of the remote controller. This is the address to which the system sends data when it communicates with the remote controller. This value can be obtained from the remote controller's configuration record. Valid values range from hex 000000000001 through hex FFFFFFFFFFFF.

***SAME:** The value does not change.

adapter-address: Specify the adapter address of the remote controller.

DSAP
Specifies the logical address of the SNA destination service access point (DSAP) to which this system sends data when it communicates with the remote controller. This address allows the controller and the system to route the data that comes from this system. The value must match the value specified on the source service access point (SSAP) parameter in the remote controller's configuration record.

***SAME:** The value does not change.

destination-service-access-point: Specify a hexadecimal value ranging from 04 through 9C, in increments of 4 (for example, 04, 08, 0C, 10) to represent the destination service access point.

SSAP
Specifies the source service access point. This is the logical address this system uses when it sends data to the remote controller. This address allows the remote controller and the system to route the data that comes from this system. It must match the value assigned to the destination service access point (DSAP) prompt in the remote controller's configuration record.

***SAME:** The value does not change.

source-service-access-point: Specify a hexadecimal value ranging from 04 through 9C, in increments of 4 (for example, 04, 08, 0C, 10) to represent the source service access point.

LANFRMRTY
Specifies the number of times a frame is transmitted if there is no acknowledgement from the remote controller in the time period specified by the LANRSPTMR parameter. This value is only used after a successful connection has been made.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-frame-retry: Specify a value ranging from 0 through 254 for the number of times a frame is transmitted before and acknowledgement is received.

LANCNNRTY
Specifies the number of times a transmission is attempted before and acknowledgement is received. This value is used at connection time (unlike the LANFRMRTY parameter, which is used *after* a connection has been made).

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-connection-retry: Specify a value ranging from 0 through 254 for the number of times the transmission is attempted before and acknowledgement is received.

LANRSPTMR
Specifies the length of time the system waits before an inoperative condition occurs on a link *after* a connection is made.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-response-timer: Specify a value ranging from 1 through 254 in 0.1-second intervals for the amount of time the system waits before an inoperative condition occurs.

LANCNNTMR
Specifies the length of time the system waits before an inoperative condition occurs on a link *at* connection time.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-connection-timer: Specify a value ranging from 1 through 254 in 0.1-second intervals for the amount of time the system waits before an inoperative condition occurs.

LANACKTMR
Specifies the length of time the system waits before sending an acknowledgement for received frames.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-acknowledgement-timer: Specify a value ranging from 1 through 254 in 0.1-second intervals, or specify 0 to indicate no delay. If 0 is specified for this parameter, 0 must also be specified for the LANACKFRQ parameter; if a non zero value is specified for this parameter, a non zero value must also be specified for the LANACKFRQ parameter.

LANINACTMR

Specifies the length of time used to determine an inactive condition for the controller.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-inactivity-timer: Specify a value ranging from 1 through 255 in 0.1-second intervals for the length of time used to determine an inactive condition for the controller.

LANACKFRQ

Specifies the maximum number of frames that are received before an acknowledgement is sent to the controller.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-acknowledgement-frequency: Specify a value ranging from 0 through 127 for the number of frames received. If 0 is specified for this parameter, 0 must also be specified for the LANACKTMR parameter; if a non zero value is specified for this parameter, a non zero value must also be specified for the LANACKTMR parameter.

LANMAXOUT

Specifies the maximum number of frames that can be sent before an acknowledgement is received from the remote system.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-maximum-outstanding-frames: Specify a value ranging from 1 through 127 for the number of frames that can be sent before and acknowledgement is received.

LANACCPTY

Specifies the priority granted to the sending system for sending frames. The larger the number, the higher the priority. This parameter is valid only when the controller is attached to an Ethernet line.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-access-priority: Specify a value ranging from 0 through 3 to indicate how soon a frame may be sent on a local area network.

LANWDWSTP

Specifies whether to reduce the number of outstanding frames that may be sent before an acknowledgement is received from the remote system during network congestion. This parameter (LAN Window Step) also indicates the number of frames that must be successfully received before the number of outstanding frames can be increased to the maximum allowable value.

***SAME:** The value does not change.

***NONE:** The number of outstanding frames is not reduced during network congestion.

LAN-number-of-frames: Specify the number of outstanding frames to be reduced before an acknowledgement is received from the remote system.

NETLVL

Specifies the level of the X.25 network used to reach this controller. The level is specified by giving the year of the standard used by the X.25 network.

***SAME:** The value does not change.

1980: The 1980 standard is used.

1984: The 1984 standard is used.

1988: The 1988 standard is used.

LINKPCL

Specifies the link level protocol used on the X.25 network to communicate with this controller.

***SAME:** The value does not change.

***QLLC:** The QLLC protocol is used.

***ELLC:** The ELLC protocol is used.

CNNPWD

Specifies, for X.25 switched virtual circuit (SVC) controllers, the password used when connecting to this controller.

***SAME:** The value does not change.

***NONE:** No password is used.

X.25-connection-password: Specify the connection password. This password for each controller can consist of any alphanumeric characters represented by the hexadecimal range from 40 through FF.

SWTLINSLCT

Specifies the method that is used to select lines from an X.25 switched line list.

***SAME:** The value does not change.

***FIRST:** The lines are selected in the order in which they are specified.

***CALC:** The system determines the value to use.

DFTPKTSIZE

Specifies the default packet size used by the X.25 network.

Element 1: Transmit Packet Size

***SAME:** The value does not change.

***LIND:** The value specified in the line description is used as the default packet size.

transmit-packet-size: Specify a default packet size for transmission. The valid values for the packet size are 64, 128, 256, 512, 1024, 2048, and 4096.

Element 2: Receive Packet Size

***SAME:** The value does not change.

***LIND:** The value specified in the line description is used as the default packet size.

***TRANSMIT:** The value specified as the default packet size for transmission is used as the default for reception.

receive-packet-size: Specify a default packet size for reception. The valid values for the packet size are 64, 128, 256, 512, 1024, 2048, and 4096.

DFTWDWSIZE

Specifies the default window size used by the X.25 network.

Element 1: Transmit Window Size

***SAME:** The value does not change.

***LIND:** The value specified in the line description is used as the default window size.

max-transmit-window-size: Specify the appropriate default window size. Valid values range from 1 through 7 for networks that use modulus 8 packet numbering. Valid values range from 1 through 15 for networks that use modulus 128 packet numbering.

Element 2: Receive Window Size

***SAME:** The value does not change.

***LIND:** The value specified in the line description is used as the default window size.

***TRANSMIT:** The value specified as the default window size for transmission is used as the default for reception.

max-receive-window-size: Specify the appropriate default window size. Valid values range from 1 through 7 for networks that use modulus 8 packet numbering. Valid values range from 1 through 15 for networks that use modulus 128 packet numbering.

USRGRPID

Specifies the closed user-group ID for contacting this X.25 switched virtual circuit (SVC) controller. Specify a value ranging from 0 through 99, as provided by the network subscription. This parameter is not valid for permanent virtual circuit (PVC) connections. It is valid only for SVC circuit outgoing call operations and is ignored for SVC incoming call connections.

***SAME:** The value does not change.

***NONE:** No user group ID is specified.

X.25-user-group-ID: Specify a value, ranging from 0 through 99, as provided by the network subscription.

RVSCRG

Specifies whether reverse charges are accepted or requested when contacting this controller.

***SAME:** The value does not change.

***NONE:** No reverse charging for network tariff billing is accepted.

***REQUEST:** Charges are requested on outgoing call request packets.

***ACCEPT:** Reverse charging for network tariff billing is accepted on incoming requests.

***BOTH:** Both incoming and outgoing requests are accepted.

X25FRMRTY

Specifies the maximum number of times a frame is sent after the response timer ends when connected to this controller.

***SAME:** The value does not change.

X.25-frame-retry: Specify a value ranging from 0 through 21 for the number of times a frame is sent.

X25RSPTMR

Specifies the time allowed to return an acknowledgement when a frame is sent while connected to this controller.

***SAME:** The value does not change.

X.25-response-timer: Specify a value ranging from 1 through 2550 in 0.1-second intervals.

X25ACKTMR

Specifies the time period to delay sending acknowledgements for received frames.

***SAME:** The value does not change.

X.25-acknowledgement-timer: Specify a value ranging from 1 through 2550 in 0.1-second intervals.

X25INACTMR

Specifies the time period used to determine an inactive condition for the controller.

***SAME:** The value does not change.

X.25-inactivity-timer: Specify a value ranging from 1 through 2550 in 0.1-second intervals, or specify 0 to indicate no timer.

USRFCL

Specifies a string of hexadecimal digits sent to the X.25 network to request additional services. The AS/400 system allows up to 218 hexadecimal characters.

***SAME:** The value does not change.

***NONE:** No additional services are needed.

user-facilities: Specify a string of hexadecimal digits sent to the network to request additional services. The AS/400 system allows up to 218 hexadecimal characters.

CPSSN

Specifies whether this advanced peer-to-peer networking (APPN) controller supports control-point to control-point (CP to CP) sessions.

***SAME:** The value does not change.

***YES:** This controller supports CP to CP sessions.

***NO:** This controller does not support CP to CP sessions.

NODETYPE
Specifies the type of APPN node this controller represents.

***SAME:** The value does not change.

***CALC:** The AS/400 system determines the type of node this controller represents.

***NETNODE:** This node is a network node in an APPN network.

***ENDNODE:** This node is an end node in an APPN network.

***LENNODE:** This node is a low-entry networking node in an APPN network.

TMSGRPNBR
Specifies the APPN transmission group number for this controller.

***SAME:** The value does not change.

***CALC:** The system determines the value to use.

APPN-transmission-group-number: Specify a value ranging from 1 through 20 for the transmission group number for this controller.

MINSWTSTS
Specifies, for the switched connection, the minimum status that APPN requires for a controller to be considered available for routing.

***SAME:** The value does not change.

***VRYONPND:** APPN will consider the controller available for routing if the status is vary on pending, varied on, or active.

***VRYON:** Indicates that APPN will consider the controller available for routing only if the status is varied on or active.

AUTOCRTDEV
Specifies which devices are automatically created.

***SAME:** The value does not change.

***ALL:** All devices that can be automatically created for this controller, except APPC devices, are automatically created.

***DEVINIT:** Only session printer and display devices started by the SNA host controller (device-initiated) are automatically created.

***NONE:** No devices are automatically created.

AUTODLTDEV
Specifies the number of minutes an automatically created device can remain idle when the last session is unbound before the device description is varied off and deleted.

***SAME:** The value does not change.

1440: The system will automatically vary off and delete the automatically-configured idle device descriptions after 1440 minutes (24 hours).

***NO:** The system will not automatically vary off and delete the automatically-configured idle device descriptions.

wait-time: Specify the number of minutes to wait before deleting the automatically-configured idle device descriptions for this controller. Valid values range from 1 to 10,000.

USRDFN1
Specifies the first of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

***LIND:** The user-defined value specified in the line description is used.

user-defined-1: Specify a value ranging from 0 through 255.

USRDFN2
Specifies the second of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

***LIND:** The user-defined value specified in the line description is used.

user-defined-2: Specify a value ranging from 0 through 255.

USRDFN3
Specifies the third of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

***LIND:** The user-defined value specified in the line description is used.

user-defined-3: Specify a value ranging from 0 through 255.

CMNRCYLMT
Specifies the number of recovery attempts made by the system before an inquiry message is sent to the system operator. Also specifies the time (in minutes) that must elapse before the system sends an inquiry message to the system operator indicating that the recovery attempt count limit is reached.

***SAME:** The value does not change.

***SYSVAL:** The recovery limits specified in the QCMNRCYLMT system value are used.

Element 1: Maximum Recovery Limit

count-limit: Specify the number of second-level recovery attempts to be made. Valid values range from 0 through 99.

Element 2: Recovery Time Interval

time-interval: Specify the time interval (in minutes) at which the specified number of second-level recoveries are attempted. Valid values range from 0 through 120. If the value specified for *count-limit* is not 0, the value 0 specifies infinite recovery.

RECONTACT

Specifies whether to send a recontact request to a remote system when the system is varied off.

Note: This parameter is only valid for X.25 and SDLC leased lines (if LINKTYPE is *X25 or *SDLC and SWITCHED is *NO).

***SAME:** The value does not change.

***YES:** A request for recontact with the remote system is sent. If this value is specified, a status of pending is shown for the remote system.

***NO:** A request for recontact with the remote system is not sent. If this value is specified, a status of inactive is shown for the remote system.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

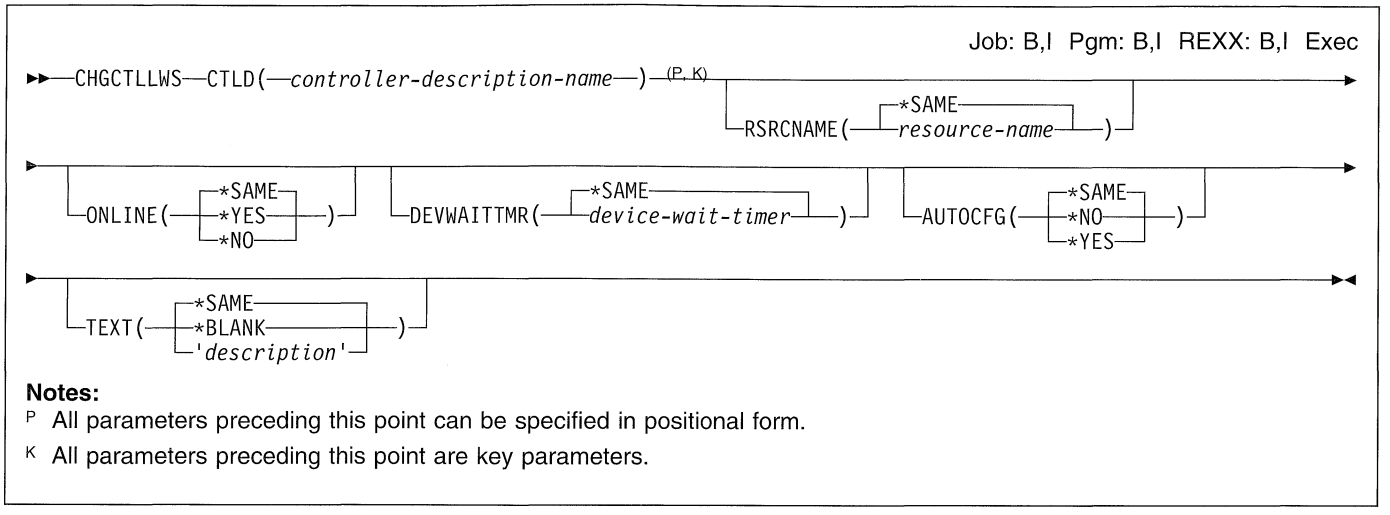
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGCTLHOST  CTLD(BOSTON)
  SSCPID(050000000011)
```

This command changes the host controller named BOSTON to have 050000000011 as its SSCPID.

CHGCTLLWS (Change Controller Description (Local Work Station)) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Controller Description (Local Work Station) (CHGCTLLWS) command changes a controller description for a local work station controller.

Required Parameter

CTLD

Specifies the name of the controller description being changed.

Optional Parameters

RSRCNAME

Specifies the resource name that describes the automatic call unit port.

***SAME:** The value does not change.

resource-name: Specify the name to identify the physical devices on the system. Use the WRKHDWRSC(*LWS) command to help determine the resource name.

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The controller is automatically varied on at IPL.

***NO:** This controller is not automatically varied on at IPL.

DEVWAITTMR

Specifies the device wait timeout value. This parameter is used to limit the amount of time a subsystem takes to complete the work station input/output. The timeout value that is used for each device is obtained from the controller to which it is attached.

A change in the DEVWAITTMR parameter value takes effect for attached devices when they are next varied on.

***SAME:** The value does not change.

device-wait-timer: Specify a value ranging from 2 through 600 that specifies the maximum number of seconds that the subsystem waits for work station input/output to be completed for all work stations attached to this controller.

When selecting a value for this parameter, the types of devices attached to the controller should be taken into account. Locally-attached work stations should have a low value for this parameter (10 seconds or less).

AUTOCFG

Specifies whether this controller description is the one that should have devices attached that have been automatically configured. Although there can be more than one controller description for each work station controller, only one controller description can be designated as an automatic device configuration controller. When new devices are automatically configured on that work station controller, they will be attached to the automatically configured controller description.

***SAME:** The value does not change.

***NO:** This is not an automatic configuration controller description.

***YES:** This is an automatic configuration controller.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

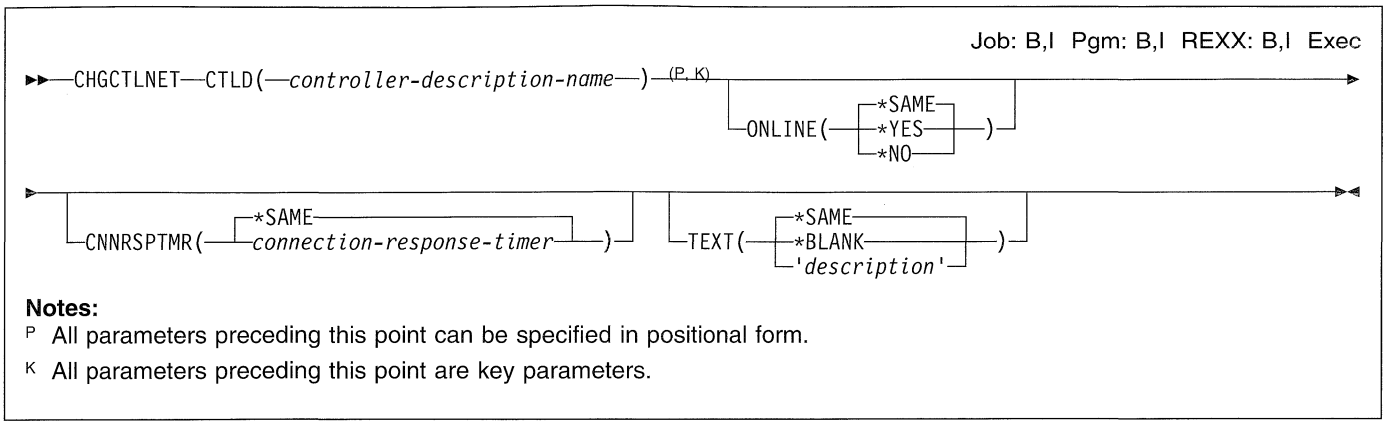
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGCTLLWS  CTLD(CTL001)  ONLINE(*NO)
```

This command changes the controller description CTL001 so that the controller is no longer varied on at initial program load (IPL).

CHGCTLNET (Change Controller Description (Network)) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Controller Description (Network) (CHGCTLNET) command changes a controller description for a network controller.

Required Parameter

CTLD

Specifies the name of the controller description being changed.

Optional Parameters

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The controller is automatically varied on at IPL.

***NO:** This controller is not automatically varied on at IPL.

CNNRSPTMR

Specifies the amount of time the system waits before responding to an incoming connection request.

***SAME:** The value does not change.

connection-response-timer: Specify the amount of time the system will wait before responding to a connection request. Valid values range from 1 through 3600 seconds.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

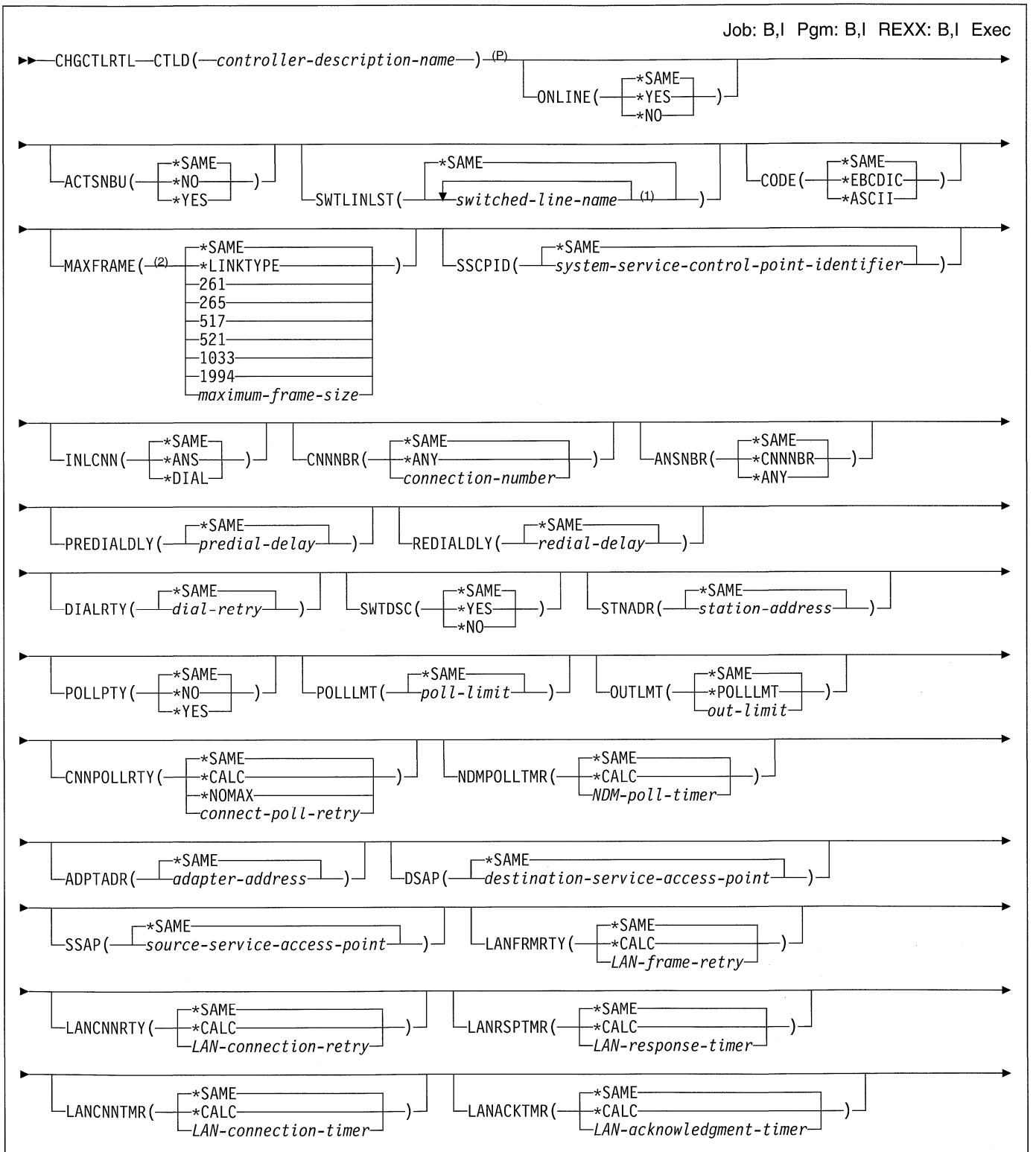
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGCTLNET   CTLD(CTL0A)  ONLINE(*NO)
```

This command changes the ONLINE parameter value to *NO for a network controller named CTL0A.

CHGCTLRTL (Change Controller Description (Retail)) Command

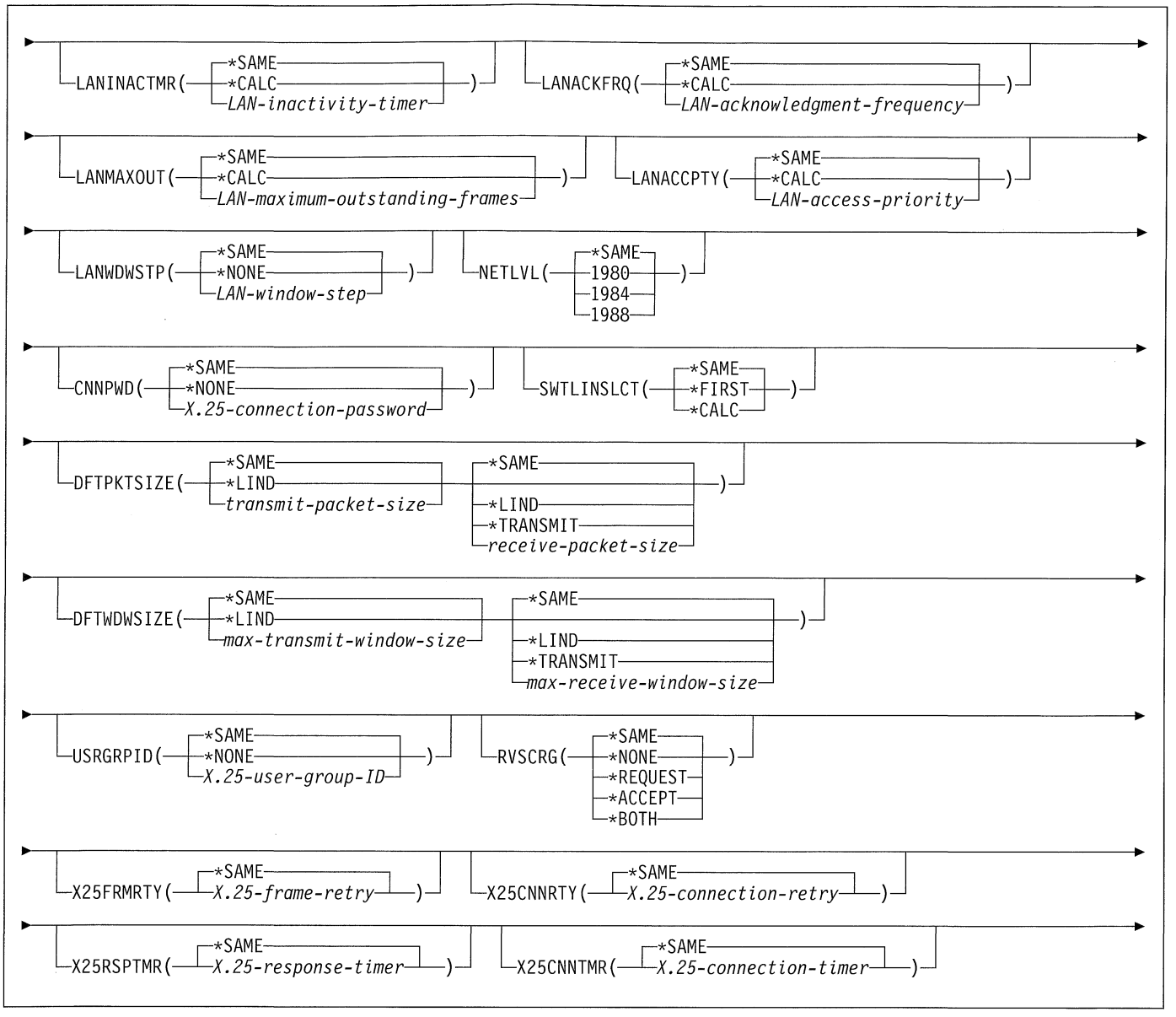


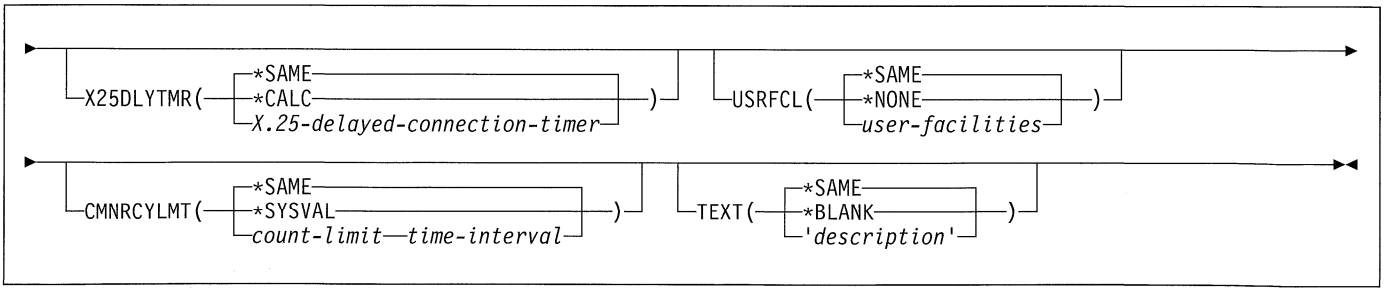
Notes:

P All parameters preceding this point can be specified in positional form.

1 A maximum of 64 repetitions

2 If the value of the LINKTYPE parameter is *LAN, values ranging from 265 through 1994 can be specified.





Purpose

The Change Controller Description (Retail) (CHGCTLRTL) command changes a controller description for a retail finance controller.

Required Parameter

CTLD

Specifies the name of the controller description being changed.

Optional Parameters

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The controller is automatically varied on at IPL.

***NO:** This controller is not automatically varied on at IPL.

ACTSNBU

Specifies, for controllers attached to nonswitched lines only, whether the switched network backup (SNBU) feature is activated or deactivated. This feature lets the user bypass a broken nonswitched connection by establishing a switched connection. This parameter applies only if SWITCHED(*NO) and SNBU(*YES) are specified when the controller description is created.

***SAME:** The value does not change.

***NO:** The switched network backup (SNBU) feature is not activated.

***YES:** The switched network backup feature (SNBU) is activated. To start the feature when the nonswitched connection is broken, specify ACTSNBU(*YES) on the line description command.

SWTLINLST

Specifies the names of the switched lines to which this controller attaches. The line descriptions must already exist. Up to 64 switched line names can be specified.

Note: The same line name can be used more than once.

***SAME:** The value does not change.

switched-line-name: Specify the name of a line for which a line description already exists. Up to 64 line names can specified.

CODE

Specifies the character code used. The code can be either extended binary-coded decimal interchange code (*EBCDIC) or the American National Standard Code for Information Interchange (*ASCII).

***SAME:** The value does not change.

***EBCDIC:** The extended binary-coded decimal interchange code (EBCDIC) character set code is used.

***ASCII:** The ASCII character set code is used.

MAXFRAME

Specifies the maximum frame (path information unit (PIU)) size that the controller can send or receive. This value is used to calculate request unit (RU) sizes. Since the maximum PIU size that the controller can send or receive is negotiated at exchange identifier time, the maximum PIU size used at run time may be different. This value matches the corresponding value on the host system.

***SAME:** The value does not change.

***LINKTYPE:** The frame size is 521 bytes for *SDLC, 1024 for *X25, and 1994 for *LAN.

maximum-frame-size: Specify either 521, 1033, or 1994 bytes as the maximum frame size for this controller. 1024 is valid only for controllers with *X25 linktype. 1994 is valid only for controllers with *LAN linktype.

SSCPID

Specifies the system service control point (SSCP) of the host system.

***SAME:** The value does not change.

system-service-control-point-identifier: Specify a 12-digit hexadecimal value for the SSCPID identifier.

INLCNN

Specifies the method used to establish a connection with this controller.

***SAME:** The value does not change.

***ANS:** The connection is made by the local system answering an incoming call from this controller.

***DIAL:** The connection is made by a call started from the local system.

For X.25, the line to which the controller attaches requires switched virtual circuits (SVCs) configured on the LGLCHLE parameter of type OUT or BOTH (*SVCOUT or *SVCBOTH) for the connection to be successful.

CNNNBR

Specifies the telephone number to dial to connect to this controller.

***SAME:** The value does not change.

***ANY:** Any valid connection number is used.

connection-number: Specify a telephone number, an X.25 network address, or an X.21 connection number depending on the type of controller and the type of line to which it is attached.

ANSNBR

Specifies the X.25 network address from which to accept calls.

***SAME:** The value does not change.

***CNNNBR:** Calls from the X.25 network address specified by the connection number (CNNNBR) parameter are accepted.

***ANY:** Calls are accepted from any X.25 network address.

PREDIALDLY

Specifies how long to wait (in 0.5 second intervals) before dialing.

***SAME:** The value does not change.

predial-delay: Specify a number ranging from 1 through 254 in 0.5-second intervals, or 0 to indicate no delay.

REDIALDLY
Specifies how long to wait (in 0.5 second intervals) before re-dialing when the call attempt is unsuccessful.

***SAME:** The value does not change.

redial-delay: Specify a value ranging from 1 through 254 in 0.5 seconds intervals, or 0 to indicate no delay.

DIALRTY
Specifies the number of re-dial attempts made by the system before considering the dialing unsuccessful.

***SAME:** The value does not change.

dial-retry: Specify a number ranging from 0 through 254 for the number of times dialing is tried.

SWTDSC
Specifies whether the switched connection to this controller is made inoperative when the last device is varied off.

***SAME:** The value does not change.

***YES:** The switched connection is disconnected when the last device is varied off.

***NO:** The switched connection is not made inoperative when the last device is varied off.

STNADR
Specifies the station address used when communicating with the controller.

***SAME:** The value does not change.

station-address: Specify the station address of the controller. Valid values range from 01 through FE.

POLLPTY
Specifies whether this controller has priority when being polled.

***SAME:** The value does not change.

***NO:** This controller does not have polling priority.

***YES:** This controller does have polling priority.

POLLLMT
Specifies the number of consecutive polls that are sent to the same controller when the poll results in receiving frames.

***SAME:** The value does not change.

poll-limit: Specify the number of consecutive polls. Valid values range from 0 through 4.

OUTLMT
Specifies the number of times SDLC allows the consecutive transmission of the maximum number of frames to a station before allowing transmission to another station.

***SAME:** The value does not change.

***POLLLMT:** The value specified on the POLLLMT parameter is used.

out-limit: Specify a value ranging from 0 through 4 for the number of consecutive transmissions.

CNNPOLLRTY
Specifies the number of times to retry connecting to a controller before reporting an error.

***SAME:** The value does not change.

***CALC:** The number of retries is 7 if the controller is switched, and *NOMAX if the controller is nonswitched.

***NOMAX:** There is no disconnect limit.

connect-poll-retry: Specify a value ranging from 0 through 65534.

NDMPOLLTMR
Specifies the interval for polling this controller when it is in normal disconnect mode (NDM). This value is specified in increments of one-tenth of a second.

***SAME:** The value does not change.

***CALC:** The system determines the value to use.

NDM-poll-timer: Specify a value ranging from 1 through 3000 in 0.1-second intervals, or 0 to indicate no timer.

ADPTADR
Specifies the 12-character adapter address of the remote controller. This is the address to which the system sends data when it communicates with the remote controller. This value can be obtained from the remote controller's configuration record. Valid values range from hex 000000000001 through hex FFFFFFFFFFFF.

***SAME:** The adapter address does not change.

adapter-address: Specify the adapter address of the remote controller.

DSAP
Specifies the logical address of the SNA destination service access point (DSAP) to which this system sends data when it communicates with the remote controller. This address allows the controller and the system to route the data that comes from this system. The value must match the value specified on the source service access point (SSAP) parameter in the remote controller's configuration record.

***SAME:** The destination service access point does not change.

destination-service-access-point: Specify a hexadecimal value ranging from 04 through 9C, in increments of 4 (for example, 04, 08, 0C, 10) to represent the destination service access point.

SSAP
Specifies the source service access point. This is the logical address this system uses when it sends data to the remote controller. This address allows the remote controller and the system to route the data that comes from this system. It must match the value assigned to the destination service access point (DSAP) prompt in the remote controller's configuration record.

***SAME:** The value does not change.

source-service-access-point: Specify a hexadecimal value ranging from 04 through 9C, in increments of 4 (for example, 04, 08, 0C, 10) to represent the source service access point.

LANFRMRTY

Specifies the number of times a frame is transmitted if there is no acknowledgment from the remote controller in the time period specified by the LANRSPTMR parameter. This value is used only after a successful connection has been made.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-frame-retry: Specify a value ranging from 0 through 254 for the number of times a frame is transmitted before an acknowledgement is received.

LANCNNRTY

Specifies the number of times a transmission is attempted before an acknowledgement is received. This value is used at connection time (unlike LANFRMRTY, which is used after a connection has been made).

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-connection-retry: Specify a value ranging from 0 through 254 for the number of times the transmission is attempted before an acknowledgement is received.

LANRSPTMR

Specifies the length of time the system waits before an inoperative condition occurs on a link after a connection is made.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-response-timer: Specify a value ranging from 1 through 254 in 0.1-second intervals for the amount of time the system waits before an inoperative condition occurs.

LANCNNTMR

Specifies the length of time the system waits before an inoperative condition occurs on a link at connection time.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-connection-timer: Specify a value ranging from 1 through 254 in 0.1-second intervals for the amount of time the system waits before an inoperative condition occurs.

LANACKTMR

Specifies the length of time the system waits before sending an acknowledgement for received frames.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-acknowledgment-timer: Specify a value ranging from 1 through 254 in 0.1-second intervals, or specify 0 to indicate no delay. If 0 is specified for this parameter, 0 must also be specified for the LANACKFRQ parameter; if a nonzero value is specified for this parameter, a nonzero value must also be specified for the LANACKFRQ parameter.

LANINACTMR

Specifies the length of time used to determine an inactive condition for the controller.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-inactivity-timer: Specify a value ranging from 1 through 255 in 0.1-second intervals for the length of time used to determine an inactive condition for the controller.

LANACKFRQ

Specifies the maximum number of frames that are received before an acknowledgment is sent to the controller.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-acknowledgment-frequency: Specify a value ranging from 0 through 127 for the number of frames received. If 0 is specified for this parameter, 0 must also be specified for the LANACKTMR parameter; if a nonzero value is specified for this parameter, a nonzero value must also be specified for the LANACKTMR parameter.

LANMAXOUT

Specifies the maximum number of frames that can be sent before an acknowledgment is received from the remote system.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-maximum-outstanding-frames: Specify a value ranging from 1 through 127 for the number of frames that can be sent before an acknowledgement is received.

LANACCPTY

Specifies the priority granted to the sending system for sending frames. The larger the number, the higher the priority.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-access-priority: Specify a value ranging from 0 through 3 to indicate how soon a frame may be sent on a local area network (LAN).

LANWDWSTP

Specifies whether to reduce to 1 the maximum number of frames outstanding to the remote system during network congestion. This parameter indicates the number of frames that must be successfully received by the remote system before the number of maximum outstanding frames can be increased by 1. The increase continues this way until the maximum number of outstanding frames reaches the value specified by the LAN maximum outstanding frames (LANMAXOUT) parameter.

***SAME:** The value does not change.

***NONE:** The number of outstanding frames is not reduced during network congestion.

LAN-window-step: Specify a value from 1 to 127 for the number of frames that must be successfully received by the remote system before the maximum number of outstanding frames can be increased by 1.

NETLVL

Specifies the level of the X.25 network used to reach this controller. The level is the year of the standard used by the/X.25 network.

***SAME:** The value does not change.

1980: The 1980 standard is used.

1984: The 1984 standard is used.

1988: The 1988 standard is used.

CNNPWD

Specifies, for X.25 switched virtual circuit (SVC) controllers, the password used when connecting to this controller.

***SAME:** The value does not change.

***NONE:** No password is used.

X.25-connection-password: Specify the connection password. The password for each controller can consist of any alphanumeric characters represented by the hexadecimal values ranging from 40 through FF.

SWTLINSLCT

Specifies the method that is used to select lines from an X.25 switched line list.

***SAME:** The value does not change.

***FIRST:** The lines are selected in the order in which they are specified.

***CALC:** The system determines the value to use.

DFTPKTSIZE

Specifies the default packet size used by the X.25 network.

Element 1: Transmit Packet Size

***SAME:** The value does not change.

***LIND:** The value specified in the line description is used as the default packet size.

transmit-packet-size: Specify a default packet size used by the X.25 network. The valid values for the packet size are 64, 128, 256, 512, 1024, 2048, and 4096.

Element 2: Receive Packet Size

***SAME:** The value does not change.

***LIND:** The value specified in the line description is used as the default packet size.

***TRANSMIT:** The value specified as the default packet size for transmission is used as the default for reception.

receive-packet-size: Specify a default packet size used by the X.25 network. The valid values for the packet size are 64, 128, 256, 512, 1024, 2048, and 4096.

DFTWDWSIZE

Specifies the default window size used by the X.25 network.

Element 1: Transmit Window Size

***SAME:** The value does not change.

***LIND:** The value specified in the line description is used as the default window size.

max-transmit-window-size: Specify the appropriate default window size. Valid values range from 1 through 7 for networks that use modulus 8 packet numbering. Valid values range from 1 through 15 for networks that use modulus 128 packet numbering.

Element 2: Receive Window Size

***SAME:** The value does not change.

***LIND:** The value specified in the line description is used as the default window size.

***TRANSMIT:** The value specified as the default window size for transmission is used as the default for reception.

max-receive-window-size: Specify the appropriate default window size. Valid values range from 1 through 7 for networks that use modulus 8 packet numbering. Valid values range from 1 through 15 for networks that use modulus 128 packet numbering.

USRGRPID

Specifies the closed user-group ID for contacting this X.25 switched virtual circuit (SVC) controller. Specify a value ranging from 0 through 99, as provided by the network subscription. This parameter is not valid for permanent virtual circuit (PVC) connections. It is valid only for SVC circuit outgoing call operations and is ignored for SVC incoming call connections.

***SAME:** The value does not change.

***NONE:** No user group identifier is specified.

X.25-user-group-ID: Specify the closed user-group ID for contacting an X.25 SVC controller. Valid values range from 0 through 99, as provided by the network subscription.

RVSCRG
Specifies whether reverse charges are accepted or requested when contacting this controller.

***SAME:** The value does not change.

***NONE:** No reverse charging for network tariff billing is accepted.

***REQUEST:** Charges are requested on outgoing call request packets.

***ACCEPT:** Reverse charging for network tariff billing is accepted on incoming requests.

***BOTH:** Both incoming and outgoing requests are accepted.

X25FRMRTY
Specifies the maximum number of times a frame is sent after the response timer ends when connected to this controller.

***SAME:** The value does not change.

X.25-frame-retry: Specify a value ranging from 0 through 21 for the number of times a frame is sent.

X25CNNRTY
Specifies the maximum number of times that a frame is sent after the connect response timer ends when connecting to this controller.

***SAME:** The value does not change.

X.25-connection-retry: Specify a value ranging from 0 through 21 for the retry value.

X25RSPTMR
Specifies the time allowed to return an acknowledgment when a frame is sent while connected to this controller.

***SAME:** The value does not change.

X.25-response-timer: Specify a value ranging from 1 through 2550 in 0.1-second intervals.

X25CNNTMR
Specifies the time allowed to return an acknowledgment when a frame is sent while connecting to this controller.

***SAME:** The value does not change.

X.25-connection-timer: Specify a value ranging from 1 through 2550 in 0.1-second intervals.

X25DLYTMR
Specifies the time interval to try establishing a connection to the controller.

***SAME:** The value does not change.

***CALC:** The system determines the value to use.

X.25-delayed-connection-timer: Specify a value ranging from 1 through 32767 in 0.1-second intervals. The system retries the connection indefinitely at the specified interval.

USRFCL
Specifies a string of hexadecimal digits sent to the X.25 network to request additional services. The AS/400 system allows up to 218 hexadecimal characters.

***SAME:** The value does not change.

***NONE:** No additional services are needed.

user-facilities: Specify a string of hexadecimal digits sent to the network to request additional services. The AS/400 system allows up to 218 hexadecimal characters.

CMNRCYLMT
Specifies the number of recovery attempts made by the system before an inquiry message is sent to the system operator. Also specifies the time (in minutes) that must elapse before the system sends an inquiry message to the system operator indicating that the recovery attempt count limit is reached.

***SAME:** The value does not change.

***SYSVAL:** The recovery limits specified in the QCMNRCYLMT system value are used.

Element 1: Maximum Recovery Limit

count-limit: Specify the number of second-level recovery attempts to be made. Valid values range from 0 through 99.

Element 2: Recovery Time Interval

time-interval: Specify the time interval (in minutes) at which the specified number of second-level recoveries are attempted. Valid values range from 0 through 120. If the value specified for *count-limit* is not 0, the value 0 specifies infinite recovery.

TEXT
Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

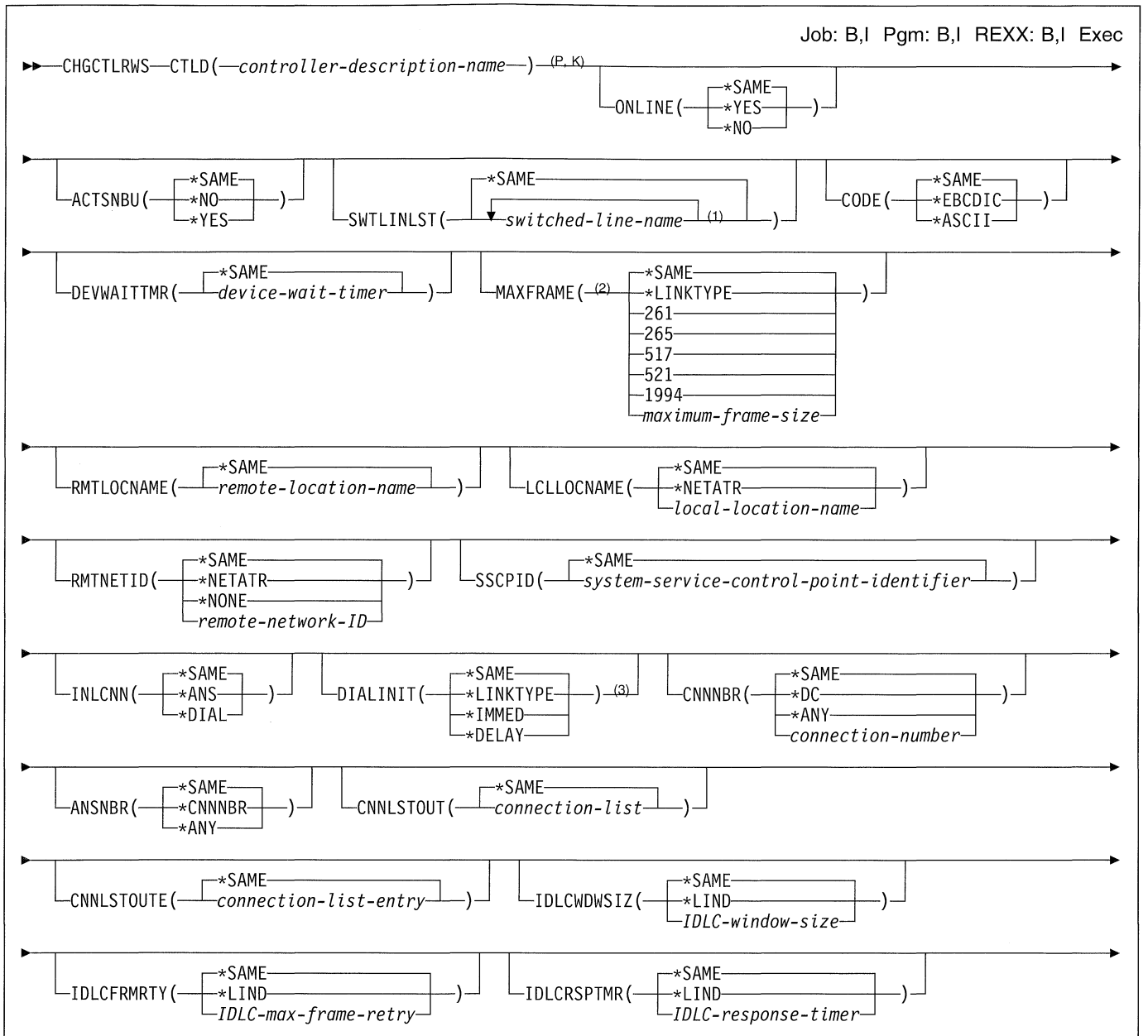
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGCTLRTL  CTLD(CTLR05)  SWTLINLST(LINE02)
  CNNNBR(255-3436)
```

This command changes the retail controller description CTLR05. The switched line list now contains only the name LINE02, and the connection number has changed to 255-3436.

CHGCTLRWS (Change Controller Description (Remote Work Station)) Command



Notes:

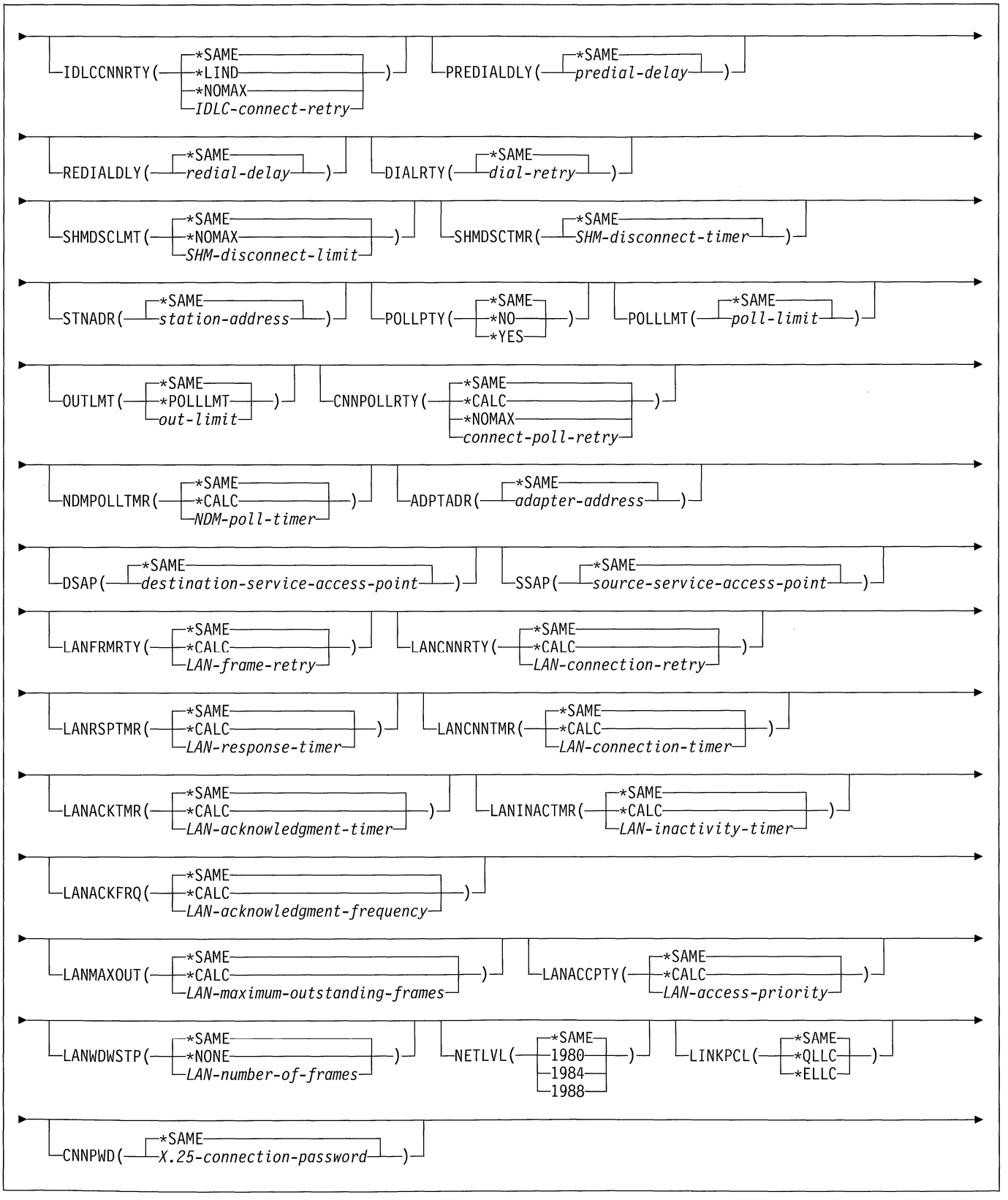
P All parameters preceding this point can be specified in positional form.

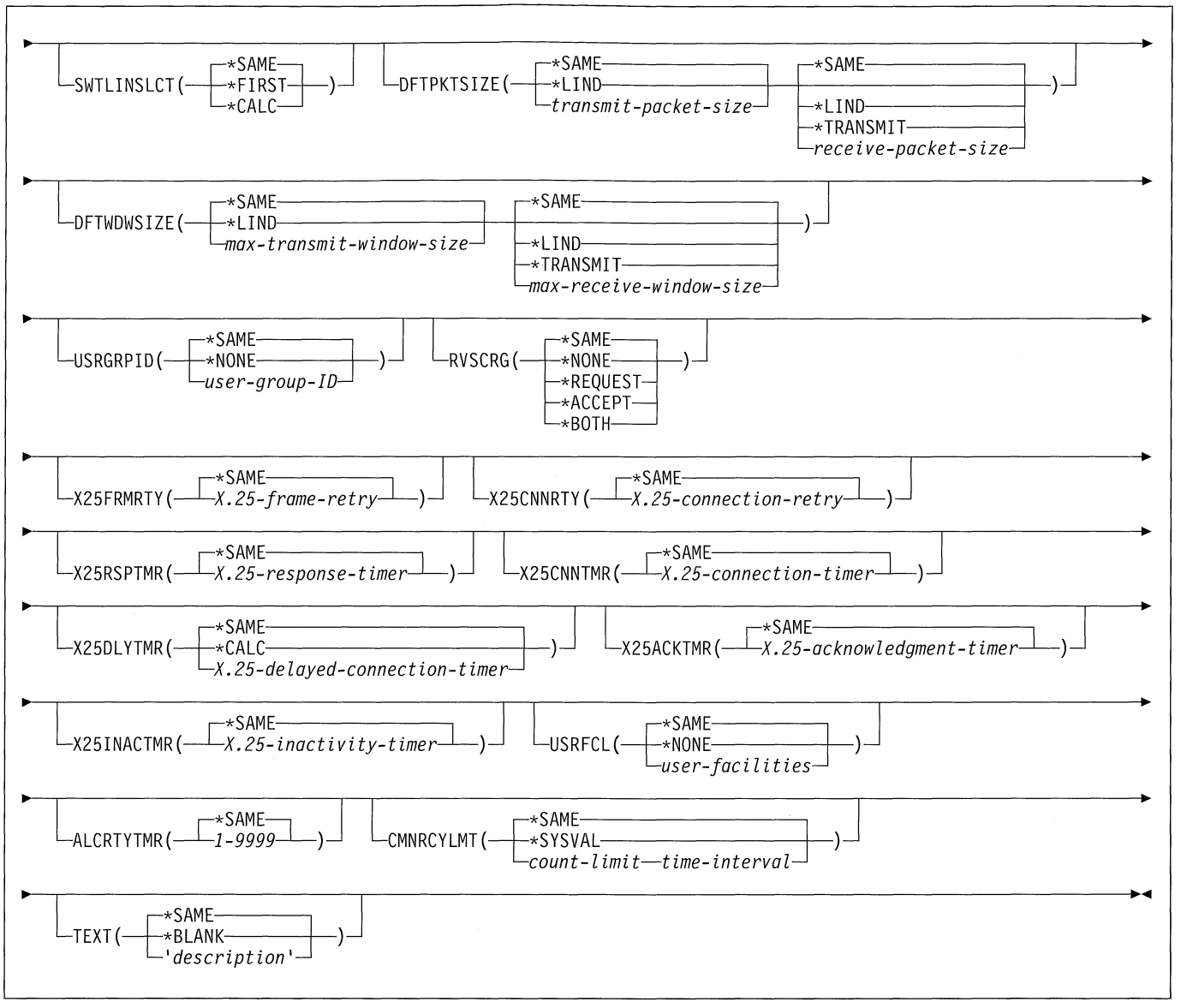
K All parameters preceding this point are key parameters.

1 A maximum of 64 repetitions

2 If the value of parameter LINKTYPE is *LAN or *TRLAN, any value ranging from 265 through 1994 can be specified.

3 This parameter is valid only if INLCNN(*DIAL) is specified.


IDLCCNNRTY(*SAME *LIND *NOMAX IDLC-connect-retry)
PREDIALDLY(*SAME predial-delay)
REDIALDLY(*SAME redial-delay)
DIALRTY(*SAME dial-retry)
SHMDSCLMT(*SAME *NOMAX SHM-disconnect-limit)
SHMDSCTMR(*SAME SHM-disconnect-timer)
STNADR(*SAME station-address)
POLLPTY(*SAME *NO *YES)
POLLLMT(*SAME poll-limit)
OUTLMT(*SAME *POLLLMT out-limit)
CNNPOLLRTY(*SAME *CALC *NOMAX connect-poll-retry)
NDMPOLLTMR(*SAME *CALC NDM-poll-timer)
ADPTADR(*SAME adapter-address)
DSAP(*SAME destination-service-access-point)
SSAP(*SAME source-service-access-point)
LANFRMRTY(*SAME *CALC LAN-frame-retry)
LANCNNRTY(*SAME *CALC LAN-connection-retry)
LANRSPTMR(*SAME *CALC LAN-response-timer)
LANCNNTMR(*SAME *CALC LAN-connection-timer)
LANACKTMR(*SAME *CALC LAN-acknowledgment-timer)
LANINACTMR(*SAME *CALC LAN-inactivity-timer)
LANACKFRQ(*SAME *CALC LAN-acknowledgment-frequency)
LANMAXOUT(*SAME *CALC LAN-maximum-outstanding-frames)
LANACCPTY(*SAME *CALC LAN-access-priority)
LANWDWSTP(*SAME *NONE LAN-number-of-frames)
NETLVL(*SAME 1980 1984 1988)
LINKPCL(*SAME *QLLC *ELLC)
CNNPWD(*SAME X.25-connection-password)



Purpose

The Change Controller Description (Remote Work Station) (CHGCTLRWS) command changes a controller description for a remote work station controller.

Required Parameter

CTLD

Specifies the name of the controller description being changed.

Optional Parameters

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** This value does not change.

***YES:** The controller is automatically varied on at IPL.

***NO:** This controller is not automatically varied on at IPL.

ACTSNBU

Specifies, for controllers attached to nonswitched lines only, whether the switched network backup (SNBU) feature is activated or deactivated. This feature lets the user bypass a broken nonswitched connection by establishing a switched connection. This parameter applies only if SWITCHED(*NO) and SNBU(*YES) are specified when the controller description is created.

***SAME:** The value does not change.

***NO:** The switched network backup (SNBU) feature is not activated.

***YES:** The switched network backup feature (SNBU) is activated. To start the feature when the nonswitched connection is broken, specify ACTSNBU(*YES) on the line description command.

SWTLINLST

Specifies the names of the switched lines to which this controller attaches. The line descriptions must already exist. Up to 64 switched line names can be specified.

Note: The same line name can be used more than once.

***SAME:** The value does not change.

switched-line-name: Specify the name of a line (64 maximum) for which a line description already exists.

CODE

Specifies the character code used. The code can be either extended binary-coded decimal interchange code (*EBCDIC) or the American National Standard Code for Information Interchange (*ASCII).

***SAME:** The value does not change.

***EBCDIC:** The extended binary-coded decimal interchange code (EBCDIC) character set code is used.

***ASCII:** The ASCII character set code is used.

DEVWAITTMR

Specifies the device wait timeout value. This is used to limit the amount of time that a subsystem takes for the work station input/output to complete. The timeout value that is used for each device is obtained from the controller to which it is attached at vary on time.

A change in the DEVWAITTMR parameter value takes effect for attached devices the next time they are varied on.

***SAME:** The value does not change.

device-wait-timer: Specify a value ranging from 2 through 600 seconds.

MAXFRAME

Specifies the maximum frame (path information unit (PIU)) size that the controller can send or receive. This value is used to calculate request unit (RU) sizes. Since the maximum PIU size that the controller can send or receive is negotiated at exchange identifier time, the maximum PIU size used at run time may be different. This value matches the corresponding value on the host system.

Note: This parameter can be specified only for 3174 and 5394 controllers.

***SAME:** The value does not change.

***LINKTYPE:** If *LINKTYPE is specified, the system uses one of the following values (sizes in bytes):

- For 3174 controllers, the following values are used:

```
*SDLC - 265    *IDLC - 521
*X25 -  256    *LAN - 1994
```

- For 5394 controllers, the following values are used:

```
*SDLC - 517
*X25  - 512
```

Note: A 5394 controller cannot be connected to local area network (*LAN).

maximum-frame-size: Specify the maximum frame size (PIU size) that can be used for this controller. The value specified for this parameter depends on the type of line and controller being used. The following values (measured in bytes) can be specified for this parameter:

- For 5394 controllers:

```
*SDLC - 261, 517
*X25  - 256, 265, 512, 521
```

- For 3174 controllers:

```
*SDLC    - 265
*X25     - 256, 265
*LAN     - 265 to 1994
*IDLC    - 265 to 1994
```

RMTLOCNAME

Specifies the name by which the controller is known to the local system.

***SAME:** The remote location name does not change.

remote-location-name: Specify a maximum of 8 characters for the remote location name.

LCLLOCNAME

Specifies the local location name.

***SAME:** The local location name does not change.

***NETATR:** The local location name defined in the network attributes is used.

local-location-name: Specify a maximum of 8 characters for the local location name.

RMTNETID

Specifies the name of the remote network in which the adjacent control point resides. In general, the network identifier must be the same for all nodes in the network.

***SAME:** The remote network identifier does not change.

***NETATR:** The local network identifier defined in the network attributes is used as the remote network identifier.

***NONE:** No remote network identifier is used.

remote-network-ID: Specify a maximum of 8 characters for the remote network identifier.

SSCPID

Specifies the system service control point (SSCP) of the host system.

***SAME:** The value does not change.

system-service-control-point-identifier: Specify a 12-digit hexadecimal value for the SSCPID identifier.

INLCNN

Specifies the method used to make the initial connection on a switched line between the system and the remote controller. For X.25 lines, this parameter specifies the

type of switched virtual circuit (SVC) connection that is made over the X.25 line.

***SAME:** The value does not change.

If a call is received from the remote controller and all necessary conditions are met, the incoming call can be answered by the system. For X.25, the line the controller attaches to has to be switched virtual circuits (SVC) configured on the LGLCHLE parameter of type IN or BOTH(*SVCIN or *SVCBOTH) for the connection to be successful.

***ANS:** The connection is made by the local system answering an incoming call from this controller.

***DIAL:** The connection is made by a call started from the local system.

For X.25, the line to which the controller attaches requires switched virtual circuits (SVCs) configured on the LGLCHLE parameter of type OUT or BOTH (*SVCOUT or *SVCBOTH) for the connection to be successful.

DIALINIT

Specifies the method used to make the initial dial on a switched line between the system and the remote controller.

***SAME:** The value does not change.

***LINKTYPE:** The type of dial connection initiated is specified on the LINKTYPE parameter. For LAN or SDLC short-hold mode connections, the default is to dial the connection immediately upon vary on of the controller description. For all other link types, the default is to defer the dial.

***IMMED:** The dial connection is initiated immediately upon vary on of the controller description.

***DELAY:** The dial connection is delayed until a job is initiated that requests the use of the remote controller resources.

CNNNBR

Specifies the telephone number to dial to connect to this controller.

***SAME:** The value does not change.

***DC:** For X.21 circuit switched connections, a direct call is used to connect to this controller.

***ANY:** Calls are accepted from any network address.

connection-number: Specify a telephone number, an X.25 network address, or an X.21 connection number, depending on the type of controller and the type of line to which it is attached.

ANSNBR

Specifies the X.25 network address from which to accept calls.

***SAME:** The value does not change.

***CNNNBR:** Calls from the X.25 network address specified by the connection number (CNNNBR) parameter are accepted.

***ANY:** Calls are accepted from any X.25 network address.

CNNLSTOUT

Specifies, for ISDN switched connections, the name of a connection list object that contains the Public Switched Network assigned numbers for a dial out operation to the Public Switched Data Network.

***SAME:** The value does not change.

connection-list: Specify the name of a connection list object.

CNNLSTOUTE

Specifies, for ISDN switched connections, the entry name from the connection list that is used to make a call to the Public Switched Data Network. The connection list must be specified on the CNNLSTOUT parameter.

***SAME:** The value does not change.

connection-list-entry: Specify an entry name.

IDLCWDWSIZ

Specifies the window size used by the controller description.

***SAME:** The value does not change.

***LIND:** The window size specified in the line description is used.

IDLC-window-size: Specify the window size. Valid values range from 1 through 31.

IDLCFRMRTY

Specifies the maximum number of attempts to transmit a frame before an error is reported.

***SAME:** The value does not change.

***LIND:** The number of attempts specified in the line description is used.

max-frame-retry: Specify the number of attempts. Valid values range from 0 through 100.

IDLCRSPTMR

Specifies the amount of time, in tenths of seconds, to wait before retransmitting a frame if acknowledgement has not been received.

***SAME:** The value does not change.

***LIND:** The time specified in the line description is used.

IDLC-response-timer: Specify the amount of time to wait. Valid values range from 10 through 100 tenths of seconds. For example, 10 seconds equals 100 tenths of a second.

IDLCCNNRTY

Specifies the number of times to attempt retransmission at connection time.

***SAME:** The value does not change.

***LIND:** The number of attempts specified in the line description is used.

***NOMAX:** There is no disconnect limit.

IDLC-connect-retry: Specify the number of attempts. Valid values range from 1 through 100.

PREDIALDLY
Specifies how long to wait (in 0.5 second intervals) before dialing.

***SAME:** The value does not change.

predial-delay: Specify a value ranging from 1 through 254 in 0.5-second intervals, or 0 to indicate no delay.

REDIALDLY
Specifies how long to wait (in 0.5 second intervals) before re-dialing when the call attempt is unsuccessful.

***SAME:** The value does not change.

redial-delay: Specify a value ranging from 1 through 254 in 0.5 seconds intervals, or 0 to indicate no delay.

DIALRTY
Specifies the number of re-dial attempts made by the system before considering the dialing unsuccessful.

***SAME:** The value does not change.

dial-retry: Specify a number ranging from 0 through 254 for the number of times the dialing is tried.

SHMDSCLMT
Specifies the number of nonproductive responses (RR or RNR) that are required from the remote station before the connection can be suspended for this X.21 short-hold mode connection. This parameter is used only if SHM(*YES) was specified when the description was created.

***SAME:** The value does not change.

***NOMAX:** There is no disconnect limit.

SHM-disconnect-limit: Specify a value ranging from 1 through 254, indicating the number of nonproductive responses that must be received before the connection can be suspended.

SHMDSCTMR
Specifies the minimum length of time that the primary system maintains the connection to the remote system for this X.21 short-hold mode controller, in tenths of a second. This parameter is used only if SHM(*YES) was specified when the description was created.

***SAME:** The value does not change.

SHM-disconnect-timer: Specify a value ranging from 2 through 3000, that indicates the minimum length of time, in tenths of a second, that the primary system maintains the connection to the remote system.

STNADR
Specifies the station address used when communicating with the controller.

***SAME:** The value does not change.

station-address: Specify the station address of the remote controller. Valid values range from 01 through FE.

POLLPTY
Specifies whether this controller should have priority when being polled.

***SAME:** The value does not change.

***NO:** This controller does not have polling priority.

***YES:** This controller has polling priority.

POLLLMT
Specifies the number of consecutive polls that are sent to the same controller when the poll results in receiving frames.

***SAME:** The value does not change.

poll-limit: Specify a value ranging from 0 through 4.

OUTLMT
Specifies the number of times SDLC allows the consecutive transmission of the maximum number of frames to a station before allowing transmission to another station.

***SAME:** The value does not change.

***POLLLMT:** The value specified on the POLLLMT parameter is used.

out-limit: Specify a value ranging from 0 through 4 for the number of consecutive transmissions.

CNNPOLLRTY
Specifies the number of times to retry connecting before reporting an error.

***SAME:** The value does not change.

***CALC:** The number of retries is 7 if the controller is switched, and *NOMAX if the controller is nonswitched.

***NOMAX:** There is no disconnect limit.

connect-poll-retry: Specify a value ranging from 0 through 65534.

NDMPOLLTMR
Specifies the interval for polling this controller when it is in normal disconnect mode (NDM). This value is specified in increments of one-tenth of a second.

***SAME:** The value does not change.

***CALC:** The system determines the value to use.

NDM-poll-timer: Specify a value ranging from 1 through 3000 in 0.1 second intervals, or 0 to indicate no timer.

ADPTADR
Specifies the 12-character adapter address of the remote controller. This is the address to which the system sends data when it communicates with the

remote controller. This value can be obtained from the remote controller's configuration record. Valid values range from hex 000000000001 through hex FFFFFFFFFFFF.

***SAME:** The value does not change.

adapter-address: Specify the adapter address of the remote controller.

DSAP

Specifies the logical address of the SNA destination service access point (DSAP) to which this system sends data when it communicates with the remote controller. This address allows the controller and the system to route the data that comes from this system. The value must match the value specified on the source service access point (SSAP) parameter in the remote controller's configuration record.

***SAME:** The value does not change.

destination-service-access-point: Specify a hexadecimal value ranging from 04 through 9C, in increments of 4 (for example, 04, 08, 0C, 10) to represent the destination service access point.

SSAP

Specifies the source service access point. This is the logical address this system uses when it sends data to the remote controller. This address allows the remote controller and the system to route the data that comes from this system. It must match the value assigned to the destination service access point (DSAP) prompt in the remote controller's configuration record.

***SAME:** The value does not change.

source-service-access-point: Specify a hexadecimal value ranging from 04 through 9C, in increments of 4 (for example, 04, 08, 0C, 10) to represent the source service access point.

LANFRMRTY

Specifies the number of times a frame is transmitted if there is no acknowledgment from the remote controller in the time period specified by the LANRSPTMR parameter. This value is used only after a successful connection has been made.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-frame-retry: Specify a value ranging from 0 through 254 for the number of times a frame is transmitted before an acknowledgement is received.

LANCNNRTY

Specifies the number of times a transmission is attempted before an acknowledgement is received. This value is used at connection time (unlike LANFRMRTY, which is used after a connection has been made).

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-connection-retry: Specify a value ranging from 0 through 254 for the number of times the transmission is attempted before an acknowledgement is received.

LANRSPTMR

Specifies the length of time the system waits before an inoperative condition occurs on a link after a connection is made.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-response-timer: Specify a value ranging from 1 through 254 in 0.1-second intervals for the amount of time the system waits before an inoperative condition occurs.

LANCNNTMR

Specifies the length of time the system waits before an inoperative condition occurs on a link at connection time.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-connection-timer: Specify a value ranging from 1 through 254 in 0.1-second intervals for the amount of time the system waits before an inoperative condition occurs.

LANACKTMR

Specifies the length of time the system waits before sending an acknowledgement for received frames.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-acknowledgment-timer: Specify a value ranging from 1 through 254 in 0.1-second intervals, or specify 0 to indicate no delay. If 0 is specified for this parameter, 0 must also be specified for the LANACKFRQ parameter; if a nonzero value is specified for this parameter, a nonzero value must also be specified for the LANACKFRQ parameter.

LANINACTMR

Specifies the length of time used to determine an inactive condition for the controller.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-inactivity-timer: Specify a value ranging from 1 through 255 in 0.1-second intervals for the length of time used to determine an inactive condition for the controller.

LANACKFRQ

Specifies the maximum number of frames that are received before an acknowledgment is sent to the controller.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-acknowledgment-frequency: Specify a value ranging from 0 through 127 for the number of frames received. If 0 is specified for this parameter, 0 must also be specified for the LANACKTMR parameter; if a nonzero value is specified for this parameter, a nonzero value must also be specified for the LANACKTMR parameter.

LANMAXOUT

Specifies the maximum number of frames that can be sent before an acknowledgment is received from the remote system.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-maximum-outstanding-frames: Specify a value ranging from 1 through 127 for the number of frames that can be sent before an acknowledgement is received.

LANACCPTY

Specifies the priority granted to the sending system for sending frames. The larger the number, the higher the priority.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

LAN-access-priority: Specify a value ranging from 0 through 3 to indicate how soon a frame may be sent on a local area network (LAN).

LANWDWSTP

Specifies whether to reduce the number of outstanding frames that may be sent before an acknowledgement is received from the remote system during network congestion. This parameter (LAN Window Step) also indicates the number of frames that must be successfully received before the number of outstanding frames can be increased to the maximum allowable value.

***SAME:** The value does not change.

***NONE:** The number of outstanding frames is not reduced during network congestion.

LAN-number-of-frames: Specify the number of outstanding frames to be reduced before an acknowledgement is received from the remote system.

NETLVL

Specifies the level of the X.25 network that is used to reach this controller. The level is specified by giving the year of the standard used by the X.25 network.

***SAME:** The value does not change.

1980: The 1980 standard is used.

1984: The 1984 standard is used.

1988: The 1988 standard is used.

LINKPCL

Specifies the link protocol being used on the X.25 network to communicate with this controller.

***SAME:** The value does not change.

***QLLC:** The Qualified Logical Link Control (QLLC) protocol is used.

***ELLC:** The Enhanced Logical Link Control (ELLC) protocol is used.

CNNPWD

Specifies the X.25 network password to be used for password exchange with the X.25 Call Request and Incoming Call packets (Call User Data Field). This parameter is not valid for permanent virtual circuits (PVCs).

Note: This connection password is highly recommended for controllers that operate with the enhanced logical link protocol LINKPCL(*ELLC) on SVC connections. This enhanced protocol supports reconnection of virtual circuits after network errors which disconnected with a clear signal, however, this reconnection is not allowed to proceed without the password validation procedure being used. This password is concatenated with the CNNNBR parameter being used as a unique identifier for the controller.

***SAME:** The value does not change.

X.25-connection-password: Specify a password up to 8 characters in length. If less than 8 characters are specified, the field is padded with blanks. The password on incoming call requests must match this password or no connection is allowed. Valid passwords consist of hexadecimal values ranging from 40 through FF. The keyboard available to the user may only support a subset of these characters.

SWTLINSLCT

Specifies the method that is used to select lines from an X.25 switched line list.

***SAME:** The value does not change.

***FIRST:** The lines are selected in the order in which they are specified.

***CALC:** The system determines the value to use.

DFTPKTSIZE

Specifies the default packet size used by the X.25 network.

Element 1: Transmit Packet Size

***SAME:** The value does not change.

***LIND:** The value specified in the line description is used as the default packet size.

transmit-packet-size: Specify a default packet size for transmission. The valid values for the packet size are 64, 128, 256, 512, 1024, 2048, and 4096.

Element 2: Receive Packet Size

***SAME:** The value does not change.

***LIND:** The value specified in the line description is used as the default packet size.

***TRANSMIT:** The value specified as the default packet size for transmission is used as the default for reception.

receive-packet-size: Specify a default packet size for reception. The valid values for the packet size are 64, 128, 256, 512, 1024, 2048, and 4096.

DFTWDWSIZE

Specifies the default window size used by the X.25 network.

Element 1: Transmit Window Size

***SAME:** The value does not change.

***LIND:** The value specified in the line description is used as the default window size.

max-transmit-window-size: Specify the appropriate default window size. Valid values range from 1 through 7 for networks that use modulus 8 packet numbering. Valid values range from 1 through 15 for networks that use modulus 128 packet numbering.

Element 2: Receive Window Size

***SAME:** The value does not change.

***LIND:** The value specified in the line description is used as the default window size.

***TRANSMIT:** The value specified as the default window size for transmission is used as the default for reception.

max-receive-window-size: Specify the appropriate default window size. Valid values range from 1 through 7 for networks that use modulus 8 packet numbering. Valid values range from 1 through 15 for networks that use modulus 128 packet numbering.

USRGRPID

Specifies the closed user-group ID for contacting this station as provided by the network subscription. This parameter is not valid for permanent virtual circuit (PVC) connections. It valid only for SVC circuits outgoing call operations and is ignored for SVC incoming call connections.

***SAME:** The value does not change.

***NONE:** No group identifier is used.

user-group-ID: Specify the 2-decimal digit identifier as provided by the network subscription.

RVSCRG

Specifies whether reverse charges are accepted or requested when contacting this controller.

***SAME:** The value does not change.

***NONE:** No reverse charging for network tariff billing is accepted.

***REQUEST:** Charges are requested on outgoing call request packets.

***ACCEPT:** Reverse charging for network tariff billing is accepted on incoming requests.

***BOTH:** Both incoming and outgoing requests are accepted.

X25FRMRTY

Specifies the maximum number of times that a frame is sent after the response timer ends when connected to this controller.

***SAME:** The value does not change.

X.25-frame-retry: Specify a value ranging from 0 through 21 for the number of tries.

X25CNNRTY

Specifies the maximum number of times that a frame is sent after the connect response timer ends when connecting to this controller.

***SAME:** The value does not change.

X.25-connection-retry: Specify a value ranging from 0 through 21 for the number of tries.

X25RSPTMR

Specifies the time allowed for a response to be returned when a frame is sent while connected to this controller.

***SAME:** The value does not change.

X.25-response-timer: Specify a value ranging from 1 through 2550 in 0.1-second intervals.

X25CNNTMR

Specifies the time allowed for a response to be returned when a frame is sent while connecting to this controller.

***SAME:** The value does not change.

X.25-connection-timer: Specify a value ranging from 1 through 2550 in 0.1-second intervals.

X25DLYTMR

Specifies how often to try establishing a connection to the controller.

***SAME:** The value does not change.

***CALC:** The AS/400 system uses the values specified for the X25CNNTMR and X25CNNRTY parameters to determine how often and how many times to try establishing the connection.

X.25-delayed-connection-timer: Specify a value ranging from 1 through 32767 in 0.1-second intervals. The system retries the connection indefinitely at the specified interval.

X25ACKTMR
Specifies the time period to delay sending responses for received frames.

***SAME:** The value does not change.

X.25-acknowledgment-timer: Specify a value ranging from 1 through 2550 in 0.1-second intervals, or 0 to indicate no timer.

X25INACTMR
Specifies the time period used to determine an inactive condition for the controller.

***SAME:** The value does not change.

X.25-inactivity-timer: Specify a value ranging from 1 through 2550 in 0.1-second intervals.

USRFCL
Specifies a string of hexadecimal digits sent to the X.25 network to request additional services. The AS/400 system allows up to 218 hexadecimal characters.

***SAME:** The value does not change.

***NONE:** No user facilities are requested.

user-facilities: Specify a string of hexadecimal digits to be sent to the X.25 network to request additional services. The AS/400 system allows up to 218 hexadecimal characters.

ALCRTYTMR
Specifies the length of time, in seconds, the system waits between attempts to establish an LU6.2 session.

***SAME:** The length of time does not change.

1-9999: Specify a length of time in seconds. Valid values range from 1 through 9999.

CMNRCYLMT
Specifies the number of recovery attempts made by the system before an inquiry message is sent to the system operator. Also specifies the time (in minutes) that must elapse before the system sends an inquiry message to the system operator indicating that the recovery attempt count limit is reached.

***SAME:** The value does not change.

***SYSVAL:** The recovery limits specified in the QCMNRCYLMT system value are used.

Element 1: Maximum Recovery Limit

count-limit: Specify the number of second-level recovery attempts to be made. Valid values range from 0 through 99.

Element 2: Recovery Time Interval

time-interval: Specify the time interval (in minutes) at which the specified number of second-level recoveries are attempted. Valid values range from 0 through 120. If the value specified for *count-limit* is not 0, the value 0 specifies infinite recovery.

TEXT
Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

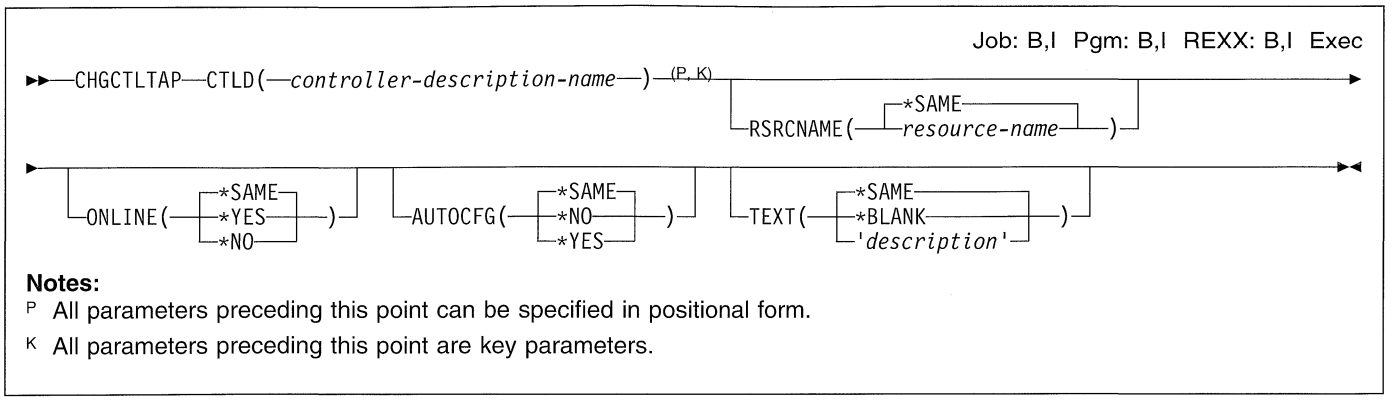
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGCTLRWS  CTLD(CTL005)  SWTLINLST(LINE01)
  CNNNBR(555-5950)
```

This command changes the controller description CTL005. The switched line list now contains only the name LINE01, and the connection number has changed to 555-5950.

CHGCTLTAP (Change Controller Description (TAPE)) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Controller Description (TAPE) (CHGCTLTAP) command changes a controller description for a tape controller.

Required Parameter

CTLD

Specifies the name of the controller description being changed.

Optional Parameters

RSRCNAME

Specifies the resource name that describes the automatic call unit port.

***SAME:** The value does not change.

resource-name: Specify the name to identify the physical devices on the system. Use the Work with Hardware Resources (WRKHDWRSC) command with *STG specified for the TYPE parameter to help determine the resource name.

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The controller is automatically varied on at IPL.

***NO:** This controller is not automatically varied on at IPL.

AUTOCFG

Specifies whether devices that have been automatically configured are being attached to this controller description. Although there can be more than one controller description for each work station controller, only one controller description can be designated as an automatic device configuration controller. When new devices are automatically configured on that work station controller, they will be attached to the automatically-configured controller description.

***SAME:** The value does not change.

***NO:** This controller description is not automatically configured.

***YES:** This controller description is automatically configured.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

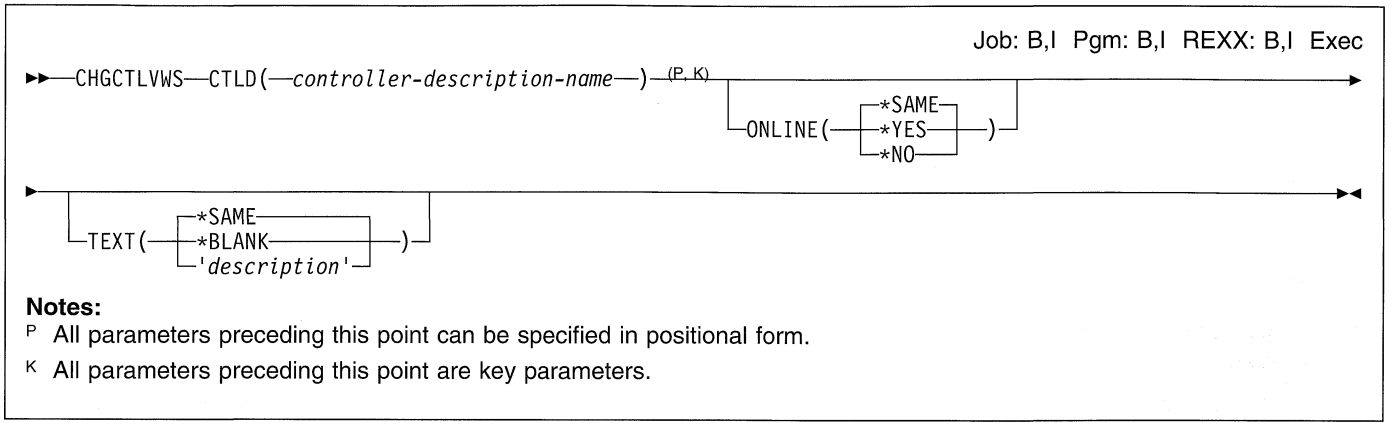
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CRTCTLTAP  CTLD(TAP01)  ONLINE(*NO)
```

This command changes the ONLINE parameter value to *NO for a tape controller named TAP01.

CHGCTLVWS (Change Controller Description (Virtual Work Station)) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Controller Description (Virtual Work Station) (CHGCTLVWS) command changes a controller description for a virtual work station (pass-through) controller.

Required Parameter

CTLD

Specifies the name of the controller description being changed.

Optional Parameters

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The controller is automatically varied on at IPL.

***NO:** This controller is not automatically varied on at IPL.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

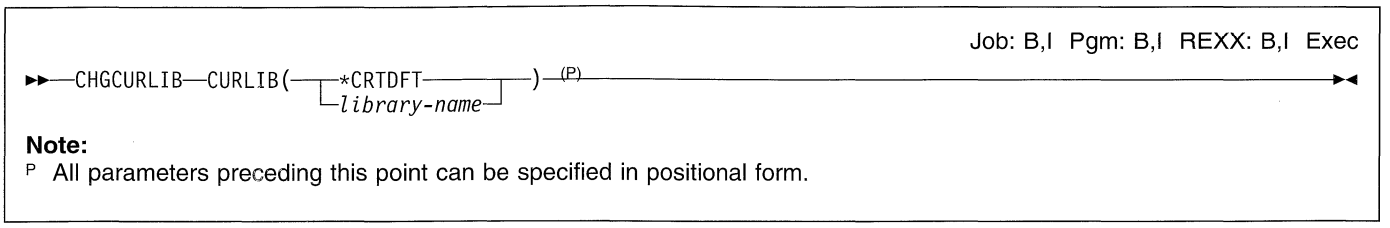
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGCTLVWS  CTLD(VRTCTL325)  TEXT('S/325 virtual
  controller')
```

This command changes the controller description of controller VRTCTL325 to have a new text description.

CHGCURLIB (Change Current Library) Command



Purpose

The Change Current Library (CHGCURLIB) command replaces the current library entry of the library list.

Note: If the current library entry of the library list is changed from a menu or program that has a current library associated with it, then the current library changes are only in effect during the call level of the menu or program from which it was changed.

Required Parameter

CURLIB

Specifies the library that replaces the current library entry in the job's library list.

***CRTDFT:** No library is the current entry of the library list. If objects are created into the current library, the QGPL library is used as the default.

library-name: Specify the name of the library that replaces the current library entry in the job's library list.

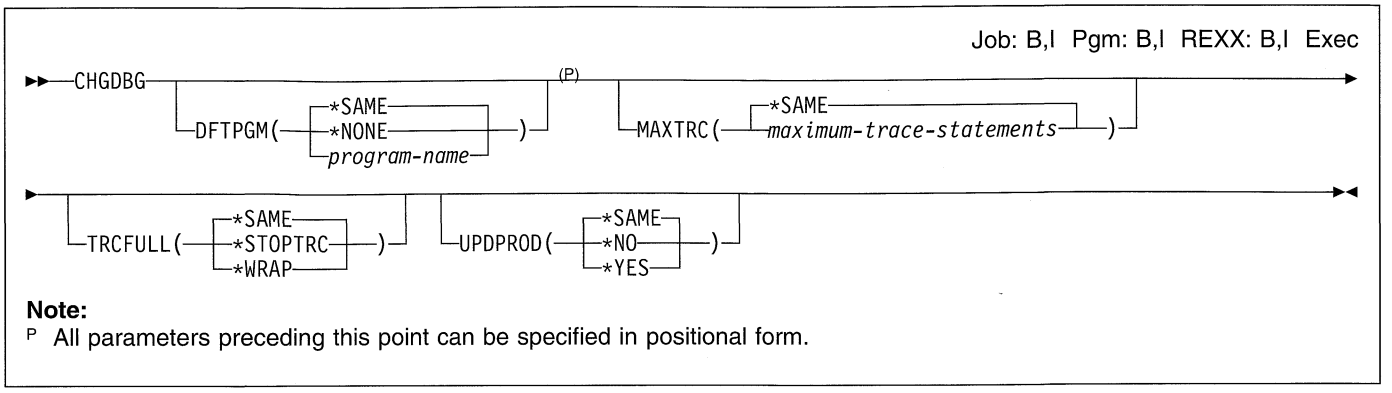
Note: QTEMP cannot be specified for this parameter.

Example

```
CHGCURLIB  CURLIB(ULIB10)
```

This command changes the current library entry of the current job's library list to contain library ULIB10.

CHGDBG (Change Debug) Command



Note:

P All parameters preceding this point can be specified in positional form.

Purpose

The Change Debug (CHGDBG) command changes the attributes of the debugging session currently in effect for a job. All of the attributes can be changed, except which programs to debug. Use the Add Program (ADDPGM) command or the Remove Program (RMVPGM) commands to add or remove a program from debug mode.

Restrictions: This command is valid only in debug mode. To start debug mode, see the STRDBG (Start Debug) command. If the user is servicing another job, and that job is ending, this command is not allowed.

Optional Parameters

DFTPGM

Specifies the name of the program to use as the default program during debug mode. The program specified here is used as the default program for any of the other debug commands that specify *DFTPGM on their PGM parameter. (That is, if a default program was previously specified, this parameter can change it.)

***SAME:** The value does not change.

***NONE:** No program is specified as the default program; if a program was specified as a default program, it is no longer the default program. If the job has no default program, *DFTPGM cannot be specified on the PGM parameter of any other debug commands.

program-name: Specify the name of the program to use as the default program during debug mode. The same name must already have been specified in the PGM parameter of the STRDBG command or ADDPGM command.

MAXTRC

Specifies the maximum number of trace statements that the system puts into the job's trace file before either stopping tracing or wrapping around (overlaying) on the trace file. When the trace file contains the maximum specified, the system performs the actions specified in the TRCFULL parameter.

Note: Instruction stepping can be performed on a program being debugged in an interactive environment by setting the maximum number of trace statements to **1** and the TRCFULL parameter to *STOPTRC.

***SAME:** The value does not change.

maximum-trace-statements: Specify the maximum number of trace statements that can be in the trace file.

TRCFULL

Specifies what happens when the job's trace file is full (that is, it contains the maximum number of trace statements specified by the MAXTRC parameter).

***SAME:** The value does not change.

***STOPTRC:** In a batch environment, tracing stops but the program continues processing. In an interactive environment, control is given to the user when a breakpoint occurs. If the user continues processing, a breakpoint occurs before processing each subsequent statement within the range of statements being traced, and the trace file is extended to contain the new entry.

***WRAP:** The trace file is overlaid with new trace statements as they occur, wrapping from the beginning of the file. The program continues processing until completed with no message to indicate that wrapping has occurred. The trace file never has more than the maximum specified statements, and they are the most recently recorded statements.

UPDPROD

Specifies whether or not database files in a production library can be opened for changes (that is, for adding, deleting, or changing records in the file) while the job is in debug mode. If not, the files must be copied into a test library before trying to run a program that uses the files.

***SAME:** The value does not change.

***NO:** Database files in production libraries cannot be changed during debug mode. Database files can be opened for reading only.

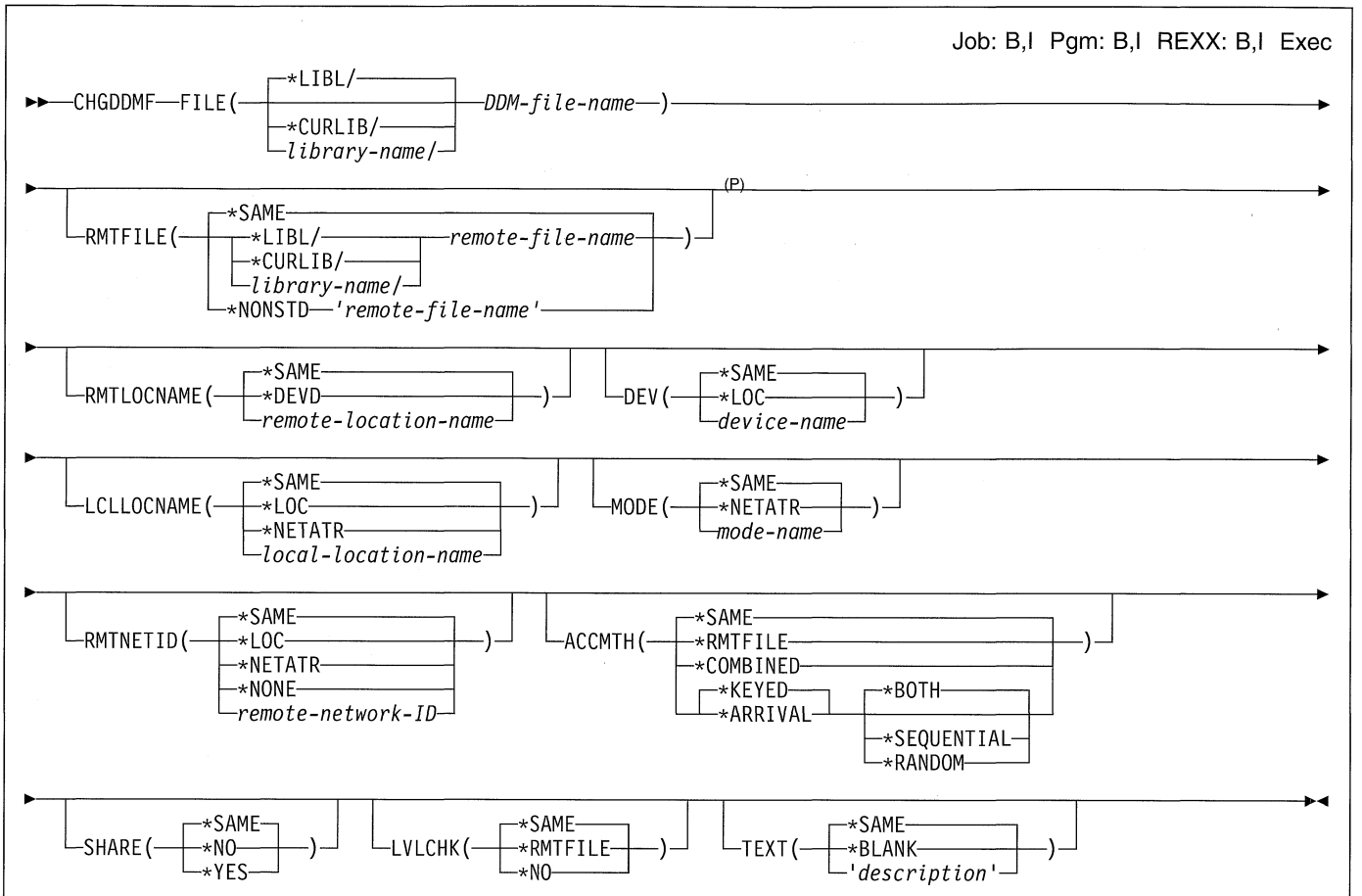
***YES:** Database files in production libraries can be changed while the job is in debug mode.

Example

```
CHGDBG   MAXTRC(400)  TRCFULL(*STOPTRC)
```

This command changes the maximum number of trace statements that can be put in the trace file to 400. The tracing is stopped when the file is full.

CHGDDMF (Change Distributed Data Management File) Command



Note:

P All parameters preceding this point can be specified in positional form.

Purpose

The Change Distributed Data Management File (CHGDDMF) command changes, in the distributed data management file (DDM) description, one or more of the attributes of the specified DDM file. The DDM file is used as a reference file by programs on the AS/400 system to access files located on any target system in the AS/400 system DDM network.

Required Parameter

FILE

Specifies the qualified name of the DDM file being changed.

The name of the DDM file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

DDM-file-name: Specify the name of the DDM file that is changed.

Optional Parameters

RMTFILE

Specifies the name of the remote file coded in the form required by the target system. This file name must be specified in code page 500. The file does not need to exist when the DDM file is created or changed.

***SAME:** The value does not change.

Element 1: Remote File Name 1

The name of the file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

Note: The library name is used only if the target system is an AS/400 system.

remote-file-name: Specify up to 10 characters for an AS/400 system file name, up to 10 characters for a System/38 simple file name, or up to 8 characters for a System/36 file name, to identify the remote file accessed. No apostrophes, blanks, or any other special characters are allowed, and any lowercase characters are always changed to uppercase.

- If the target system is an AS/400 system:
  - If *LIBL (the default library qualifier) is specified or assumed, the library list in the evoked job on the target system is used to search for the file.
  - If *CURLIB is specified, the current library in the evoked job on the target system is searched for the file.
  - A member name can be specified as part of the remote file name, but it is considered a *non-standard* name and the library, file, and member name parts must follow the value *NONSTD.
- If the target system is a System/38:
  - A qualified file name can be specified as part of the remote file name, but it is considered a non-standard name and the full file name must follow the value *NONSTD.
  - A qualified file name and member name can be specified as part of the remote file name but is considered a nonstandard name and the full file name must follow the value *NONSTD.
  - If *LIBL is specifies as the library value, the library list in the called job on the target system is searched to locate the file.
- If the target system is a System/36, the remote file name is the same as its System/36 file label, as used by System/36 OCL.

For examples of *standard* remote file names, refer to the description of this parameter in the Create Distributed Data Management File (CRTDDMF) command description.

Element 2: Remote File Name 2

***NONSTD 'remote-file-name':** For target systems that allow naming conventions other than those used by an AS/400 system and System/36, and when specifying a qualified System/38 file name, and when specifying a *member* name of a remote AS/400 system or System/38 file, enter the value *NONSTD followed by up to 255 characters for the name of the remote file accessed; the name must be coded in the form required by the target system. The name must be enclosed in apostrophes, and may contain lowercase letters, blanks, periods, or any other special characters. The AS/400 system and System/38 name must be in uppercase (because they are not changed to uppercase if enclosed in apostrophes) and no blanks are allowed.

If the target system is an AS/400 system or System/38, a file name, library name, and member name can be specified. If a member name is specified, the full file name must be enclosed in apostrophes and follow the value *NONSTD, and the member name *must* be enclosed in parentheses and immediately follow (with no space) either the library name or the file name.

For examples of specifying *nonstandard* remote file names and member names, refer to the description of this parameter in the CRTDDMF command description.

RMTLOCNAME

Specifies the name of the remote location that is used with this object.

Note: Several DDM files can use the same remote location for the target system.

***SAME:** The value does not change.

***DEVD:** The remote location name associated with the APPC device description is used.

Note: If DEV(*LOC) is specified for the APPC device description, a remote location name must be specified for RMTLOCNAME.

remote-location-name: Specify the name of the remote location that is associated with the target system. The remote location name, which is used in accessing the target system, does not need to exist when the DDM file is created or changed, but must exist when the DDM file is opened.

DEV

Specifies the name of the APPC device description on the source system that is used with this DDM file. The device description does not need to exist when the DDM file is created or changed.

More information on device names is in the *APPC Programmer's Guide*.

***SAME:** The value does not change.

***LOC:** The device associated with the remote location is used. If several devices are associated with the remote location, the system determines which device is used.

Note: If RMTLOCNAME(*DEVD) is specified for the remote location, a device name must be specified for DEV.

device-name: Specify the name of a communications device associated with the remote location. If the device name is not valid for the remote location, a message is sent when the program device entry is acquired. More

information on device names is in the *APPC Programmer's Guide*.

LCLLOCNAME

Specifies the local location name.

***SAME:** The value does not change.

***LOC:** The local location name associated with the remote location is used.

***NETATR:** The LCLLOCNAME value specified in the system network attributes is used.

local-location-name: Specify the name of the local location that is associated with the remote location. The local location name is specified only if the user wants to indicate a specific local location for the remote location. If the local location name is not valid for the remote location, an escape message is sent when the DDM file is opened.

MODE

Specifies the mode name that is used with the remote location name to communicate with the target system.

More information on mode names is in the *APPC Programmer's Guide*.

***SAME:** The value does not change.

***NETATR:** The mode name specified in the network attributes is used.

mode-name: Specify the name of the mode that is used. If the mode name is not valid for any combination of remote location and local location, an escape message is sent when the DDM file is opened.

RMTNETID

Specifies the remote network ID in which the remote location resides and is used to communicate with the target system.

More information on remote network IDs is in the *APPC Programmer's Guide*.

***SAME:** The value does not change.

***LOC:** The remote network identifier (ID) associated with the remote location is used. If several remote network IDs are associated with the remote location, the system determines which remote network ID is used.

***NETATR:** The RMTNETID value specified in the system network attributes is used.

***NONE:** No remote network identifier (ID) is used.

remote-network-ID: Specify the remote network ID that is associated with the remote location. The remote network ID is specified only if the user indicates a specific remote network ID for the remote location. If the remote network ID is not valid for the remote location, an escape message is sent when the DDM file is opened.

ACCMTH

Specifies, when the remote file is on a non-AS/400 system and non-System/38 target system, the DDM access method used to open the remote file and access its records. Specifying a value other than *RMTFILE for this parameter may improve performance when processing requests to remote files on non-AS/400 system and non-System/38 targets.

This parameter is ignored when the target system is an AS/400 system or a System/38; a remote AS/400 system or System/38 file is accessed as if it were a local file.

***SAME:** The value does not change.

***RMTFILE:** The source system selects the access method that is compatible with (a) the attributes of the remote file identified by the RMTFILE parameter and (b) the access methods used by the target system for that file. If this value is used for non-AS/400 system and non-System/38 target systems, and the source system cannot select an access method when the file is opened, a message is sent to the program user. A different value should be specified for this parameter after the target system operator has been contacted about the appropriate access method information for the file.

***COMBINED:** The DDM combined access method is used for the remote file. This access method combines the file processing capabilities of the *combined by key* (*KEYED *BOTH) and the *combined by record number* (*ARRIVAL *BOTH) access methods, as shown in the following table. The record can be selected with a key value or a record number. From that position, the position can be set relatively or randomly by key value or by record number. If duplicate keys are present in the file, they are processed in the order defined by each target system's implementation of the DDM architecture.

access-method: Specify a set of two values that indicates the access method that is used to access the remote file. If only the first value is specified (*ARRIVAL or *KEYED), the default for the second value is *BOTH, and either random or relative (sequential) selection can be requested.

The following table shows the other possible values for the ACCMTH parameter. The remote file attributes (in the far left column) refer to the type of file on the target system. The local access method (in the last three columns) refers to the way in which the source AS/400 system or System/38 program intends to access the records in the remote file.

Remote File Attributes	Local Access Method		
	***SEQUENTIAL**	***RANDOM**	***BOTH**
***ARRIVAL**	Relative by record number	Random by record number	Combined by record number
***KEYED**	Relative by key	Random by key	Combined by key

Relative by record number access method (*ARRIVAL *SEQUENTIAL): This method allows access to records relative to the current position in

record number sequence. The record number is not specified to identify the record.

Random by record number access method (*ARRIVAL *RANDOM): This method allows access to records by specifying a record number in a random sequence determined by the requester.

Combined by record number access method (*ARRIVAL *BOTH): This method combines the capabilities of the relative by record number and random by record number access methods.

Relative by key access method (*KEYED *SEQUENTIAL): This method allows records in a keyed file accessed in key value sequence. Records can be accessed by moving forward or backwards in key sequence from the current record. The key value is not specified to identify the record.

Random by key access method (*KEYED *RANDOM): This method allows records in a keyed file accessed in a random sequence. Records are selected by their key value and not their position in the file.

Combined by key access method (*KEYED *BOTH): This method combines the capabilities of the relative by key and random by key access methods.

SHARE

Specifies whether the open data path (ODP) for the distributed data management file is shared with other programs in the routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

More information on shared database files is in the *Database Guide*.

***SAME:** The value does not change.

***NO:** The ODP created by the program with this attribute is not shared with other programs in the routing step. Every time a program opens the file with this attribute, a new ODP to the file is created and activated.

***YES:** The ODP created with this attribute is shared with each program in the routing step that also specifies SHARE(*YES) when it opens the file.

Note: When SHARE(*YES) is specified and control is passed to a program, a read operation in that program retrieves the next input record. A write operation produces the next output record.

Operation considerations (regarding buffers and file position, for example) for SHARE(*YES) are the same as for database files. When an ODP is shared, the programs accessing the file share things, such as the file status and the buffer.

LVLCHK

Specifies whether the record format level identifiers in the program are checked against those in the remote file when the DDM file is opened. If so, the record format identifiers in the program must match those in the remote file. If they do not match, an error message is sent to the requesting program and neither the DDM file nor the associated remote file is opened. Files that have an error while being opened are automatically closed. This parameter can be overridden by an Override with Database File (OVRDBF) command before the remote file is opened.

***SAME:** The value does not change.

***RMTFILE:** The level identifiers of the record formats of the remote file (identified in the RMTFILE parameter) are checked at the time the DDM file is opened.

If the target system is not an AS/400 system or System/38, then the source AS/400 system creates a level check value based on the record length of the remote file and any key fields used in it. The values are then compared to the values in the program, and they must match before the remote file can be opened. This reduces the possibility that a program will select the wrong file.

Note: Before this can be done for a system other than an AS/400 system or System/38, the program must be compiled (or recompiled) using the DDM file. During the compilation, the DDM file is used to establish communications with the target system, get the remote file's attributes from the target system, and create the level identifier values so they can be included in the compiled program for later level checking.

***NO:** The level identifiers are not checked when the file is opened.

TEXT

Specifies text that briefly describes the DDM file. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

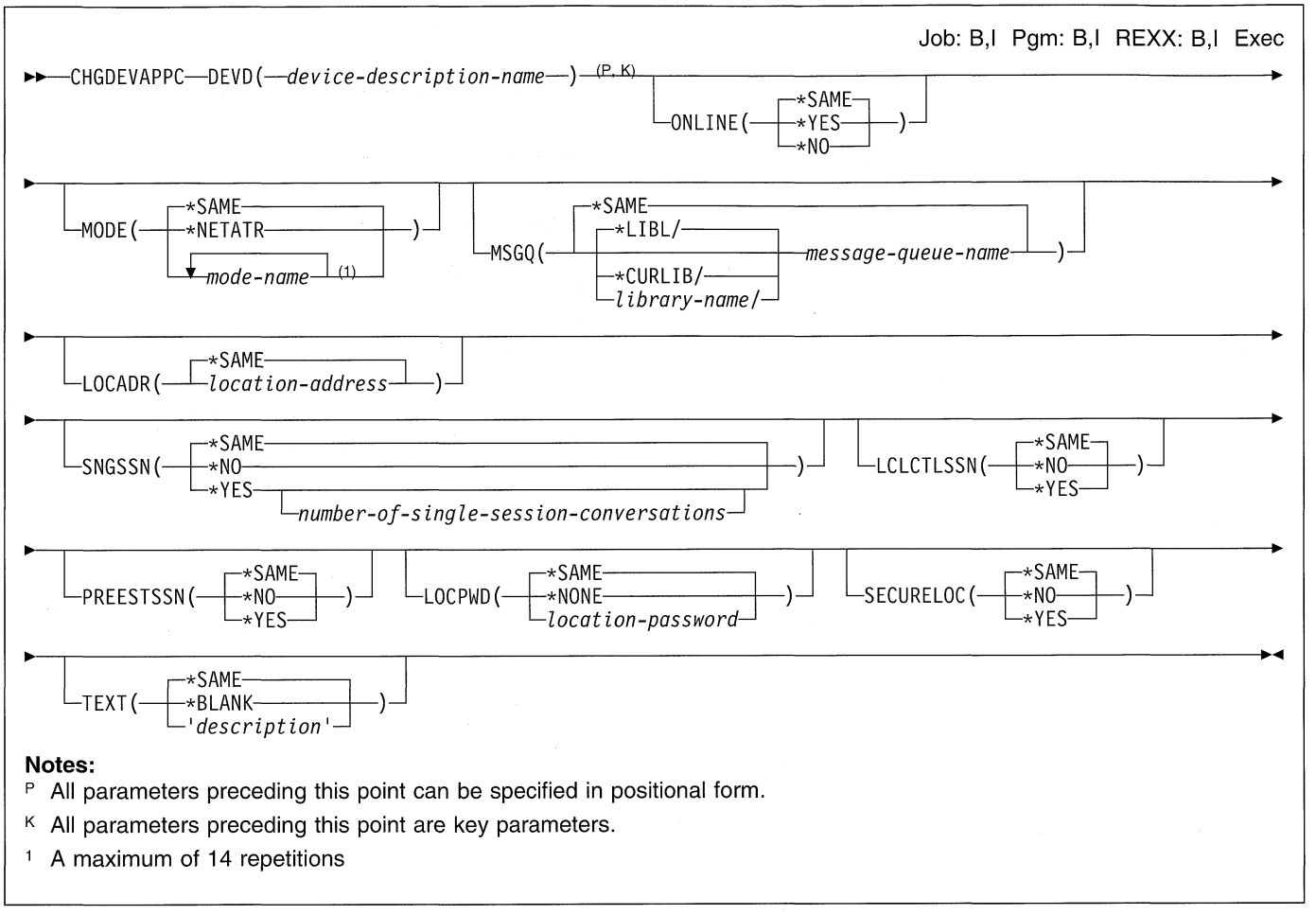
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGDDMF  FILE(SOURCE/SALES)  MODE(MODEX)
```

This command changes the communications mode for the DDM file named SALES stored in the SOURCE library on the source system. The mode is changed to MODEX.

CHGDEVAPPC (Change Device Description (APPC)) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

1 A maximum of 14 repetitions

Purpose

The Change Device Description (APPC) (CHGDEVAPPC) command changes the description of an advanced program-to-program communications (APPC) device.

Required Parameter

DEVD

Specifies the name of the device description being changed.

Optional Parameters

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The device is automatically varied on at IPL.

***NO:** This device is not automatically varied on at IPL.

MODE

Specifies the names of the modes that define the sessions on this device.

***SAME:** The value does not change.

***NETATR:** The mode name specified in the network attributes is used.

mode-name: Specify the mode name. The mode name cannot be CPSVCMG or SNASVCMG; these mode names are reserved for system use. A maximum of 14 mode names can be specified. When APPN(*YES) and LOCADR(00) are specified on the device, no modes are added to the device description. APPN automatically adds the modes.

The name can contain characters A through Z, 0 through 9, $, #, or @, but the first character cannot be 0 through 9.

MSGQ

Specifies the qualified name of the message queue to which messages are sent.

***SAME:** The message queue does not change.

The name of the message queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

message-queue-name: Specify the name of the message queue to which operational messages are sent.

LOCADR

Specifies the local location address. Specify a 2-character hexadecimal value ranging from 00 through FF.

***SAME:** The value does not change.

location-address: Specify a 2-hexadecimal character ranging from 00 through FF.

SNGSSN

Specifies whether a single session using this device description is used between the local and remote locations.

***SAME:** The value does not change.

***NO:** Multiple sessions are used on this device. LOCADR(00) is specified if SNGSSN(*NO) is specified.

Element 1: Single-Session Device Description

***YES:** This device description uses single sessions.

Element 2: Number of Single-Session Conversations

number-of-single-session-conversations: Specify the number of conversations allowed for a single session of this device description. Valid values range from 1 through 512.

LCLCTLSSN

Specifies whether a single session is locally or remotely controlled. The value of *YES is valid only if SNGSSN parameter was specified as *YES.

***SAME:** The value does not change.

***NO:** The single session is remotely controlled.

***YES:** The single session is locally controlled.

PREESTSSN

Specifies whether a single session is automatically established when the mode is started with the remote location. The value of *YES is valid only if the SNGSSN(*YES) and LCLCTLSSN(*YES) are both specified.

***SAME:** The value does not change.

***NO:** The single session is not automatically established when the mode is started.

***YES:** The single session is automatically established when the mode is started.

LOCPWD

Specifies the local password used to validate the session establishment request. The password is specified by using hexadecimal digits.

***SAME:** The value does not change.

***NONE:** No password is used to validate the session establishment request.

location-password: Specify the password to validate an even number of hexadecimal characters. Up to 16 hexadecimal characters can be specified.

For example, a password consisting of the letter A would be input as C1, and an integer 1 would be input as F1 on the LOCPWD parameter.

SECURELOC

Specifies whether the local location allows the remote location to verify user passwords and to send an already verified indicator with the program start request.

Note: Changes to this parameter are not implemented until the device is varied off and varied back on.

***SAME:** The value does not change.

***NO:** Security validation by the remote location is not accepted.

***YES:** Security validation by the remote location is accepted.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

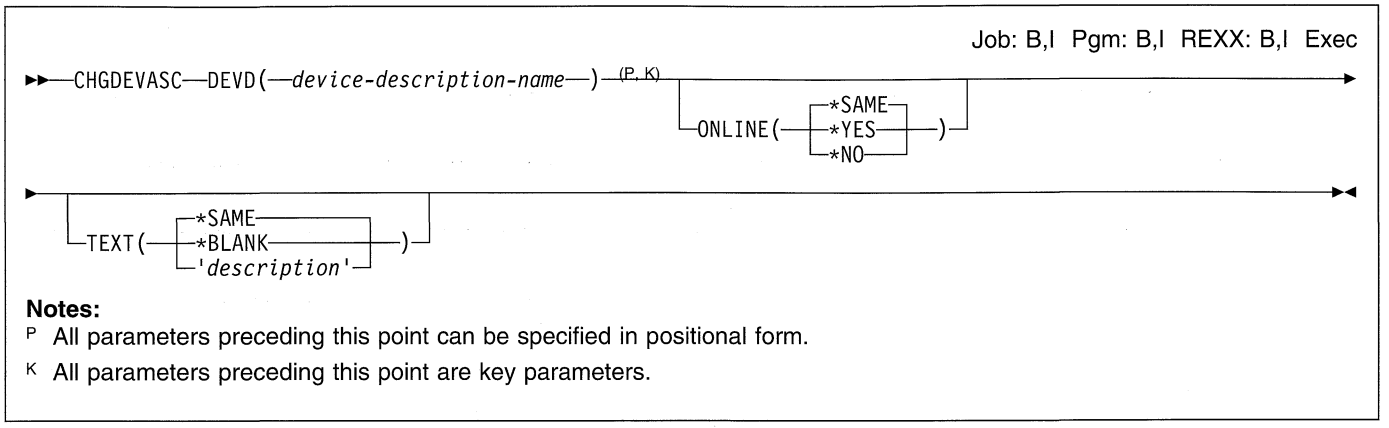
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGDEVAPPC   DEVD(APPC1)  SNGSSN(*NO)
```

This command changes the device description for communication device APPC1 so it is no longer limited to single sessions.

CHGDEVASC (Change Device Description (Async)) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Device Description (Async) (CHGDEVASC) command changes the description of an asynchronous device.

Required Parameter

DEVD

Specifies the name of the device description being changed.

Optional Parameters

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The device is automatically varied on at IPL.

***NO:** This device is not automatically varied on at IPL.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

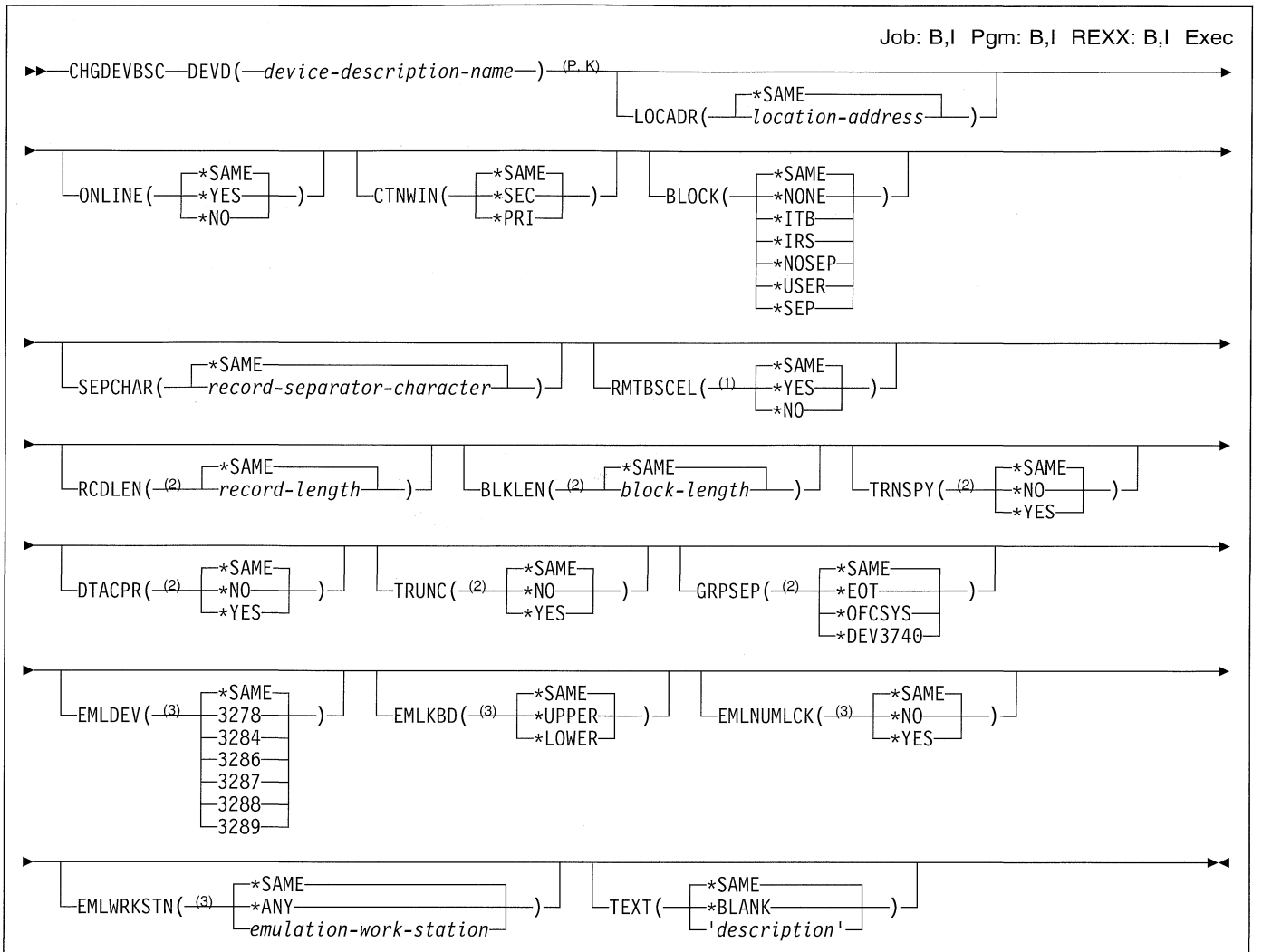
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGDEVASC   DEVD(ASC003)  ONLINE(*YES)
```

This command changes the device description for asynchronous device ASC003 so it is automatically varied on at IPL.

CHGDEVBSC (Change Device Description (BSC)) Command



Notes:

1 RMTBSCEL(*NO) must be used for *RPGT bisynchronous devices (specified as APPTYPE(*RPGT) on the CRTDEVBSC command).

2 This parameter is valid only for *BSCEL and *RPGT bisynchronous devices (specified as APPTYPE(*RPGT) or APPTYPE(*BSCEL) on the CRTDEVBSC command).

3 This parameter is valid only for *EML bisynchronous devices (specified as APPTYPE(*EML) on the CRTDEVBSC command).

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Device Description (BSC) (CHGDEVBSC) command changes the description of a binary synchronous communications (BSC) device.

Required Parameter

DEVD

Specifies the name of the device description being changed.

Optional Parameters

LOCADR

Specifies the local location address. Specify a 2-character hexadecimal value ranging from 00 through FE.

***SAME:** The value does not change.

location-address: Specify two hexadecimal characters ranging from 00 through FE. A value of 00 is used for

point-to-point switched or nonswitched connections. For multipoint tributary applications, RJE operations, or 3270 emulation, specify the unit address expected by the host system.

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The device is automatically varied on at IPL.

***NO:** This device is not automatically varied on at IPL.

CTNWIN

Specifies which BSC station in the local AS/400 system is primary and which is secondary, to resolve contention for the allocation of BSC point-to-point and multipoint lines.

***SAME:** The value does not change.

***SEC:** The local AS/400 system is the secondary station. It stops sending or receiving when the primary station calls for the line.

***PRI:** The local AS/400 system is the primary station. It has BSC line priority over the secondary BSC system.

BLOCK

Specifies whether the system or the user blocks and deblocks transmitted records. This parameter is valid only if APPTYPE(*BSCEL) or APPTYPE(*RPGT) is specified. With this parameter, the user may specify one of the following conditions of record formatting:

No blocking/deblocking: The record format described in the data description specification (DDS) is the format for both the record and the block.

User blocking/deblocking: The user must provide the BSC controls needed to describe the record format to the system.

System blocking of fixed-length records: The system uses fixed-length records, and blocks or deblocks records accordingly. The record separator character is added when a record is transmitted, and removed before the record is returned to the program. This occurs for every case except *user blocking or deblocking*.

If a parameter value other than *NONE or *USER is specified, records are blocked as required by the system for output and deblocked on input. Blocking may be done with or without record separator characters. If TRNSPY(*YES) is specified, the records may be blocked without record separator characters by specifying BLOCK(*NOSEP), or the records may be transmitted one record at a time by specifying BLOCK(*NONE). If BLOCK(*USER) is specified, records are blocked to include the BSC transparency controls. If TRNSPY(*NO) is specified, all blocking options are valid. The record length, when used, is obtained from the device file. Up to 512 records are blocked for transmission. When the system blocks or deblocks the records, record separator characters and control characters are not passed to the program as data.

***SAME:** The value does not change.

***NONE:** No blocking or deblocking is done by the system.

***ITB:** The records are blocked or deblocked, based on the location of an intermediate text block (ITB) control character. For input files, a record is delimited by locating the next ITB character, an end of text (ETX), or end-of-transmission block (ETB) character. For output files, an ITB character is inserted following the record. If the ITB is the last character of the block, it is replaced by an ETX character or an ETB character.

***IRS:** The records are blocked or deblocked, based on the location of an interrecord separator (IRS) character. For input files, a record is delimited by locating the next IRS character. For output files, an IRS character is inserted following the record.

***NOSEP:** No record separator character is in transmission blocks sent to or received from the device. The system blocks and deblocks the records to a fixed record length, as specified in the DDS format specifications, or the record length specified in the Report Generator II (RPGII) telecommunications program.

***USER:** The program provides all the control characters (including record separator characters, binary synchronous communications (BSC) framing characters, and transparency characters) necessary to send records. More information about the device and binary synchronous communications equivalence link (BSCEL) support characteristics is in the *BSC Equivalence Link Programmer's Guide*.

***SEP:** The records are blocked or deblocked based on the location of a user-specified record separator character. For input files, a record is delimited by locating the next record separator character. For output files, a record separator character is inserted after the record.

If this option is selected, an option on the SEPCHAR parameter must also be specified.

SEPCHAR

Specifies a unique one-byte record separator character. The record separator character must be specified as two hexadecimal characters. This parameter is valid only if APPTYPE(*BSCEL) or APPTYPE(*RPGT) is specified.

***SAME:** The value does not change.

record-separator-character: Specify the record separator character. The following is a list of BSC control characters that must not be used as record separator characters because they have special meanings.

EBCDIC	BSC Control
X'01'	SOH (Start of header)
X'02'	STX (Start of text)
X'03'	EXT (End of text)
X'10'	DLE (Data link escape)

X'1D' IGS (Interchange group separator)
X'1F' ITB (Intermediate text block)
X'26' ETB (End-of-transmission block)
X'2D' ENQ (Enquiry)
X'32' SYN (Synchronization)
X'37' EOT (End of transmission)
X'3D' NAK (Negative acknowledgment)

RMTBSCEL
Specifies the type of BSCEL session with the remote system. This parameter is valid only if APPTYPE(*BSCEL) is specified.

***SAME:** The value does not change.

***YES:** The remote system recognizes BSCEL start and end commands and BSCEL online messages.

***NO:** The remote system or device does not recognize BSCEL commands and BSCEL online messages, but OS/400-ICF support is desired.

Note: The value *NO must be used if APPTYPE(*RPGT) is specified.

RCDLEN
Specifies the maximum record length allowed when communicating with this device. The maximum value for this parameter is 8192 bytes. The value must not exceed the block length value for this device and must not exceed the buffer size specified on the line description (MAXBUFFER) to which this device is attached. This parameter is valid only if APPTYPE(*BSCEL) or APPTYPE(*RPGT) is specified.

***SAME:** The value does not change.

record-length: Specify a value (in bytes) that is at least the size of the largest record to be sent. Valid values range from 1 through 8192.

BLKLEN
Specifies the maximum block length allowed when communicating with this device. The maximum value for this parameter is 8192 bytes. This value must not exceed the buffer size specified on the line description (MAXBUFFER) to which this device is attached. This parameter is valid only if APPTYPE(*BSCEL) or APPTYPE(*RPGT) is specified.

***SAME:** The value does not change.

block-length: Specify the maximum block length (in bytes) of records sent. The value must be at least the size of the largest record sent. Valid values range from 1 through 8192.

TRNSPY
Specifies whether text transparency is used when sending records. Text transparency permits the transmission of all 256 EBCDIC character codes. This is used for transmitting packed or binary data fields. This parameter is valid only if APPTYPE(*BSCEL) or APPTYPE(*RPGT) is specified.

***SAME:** The value does not change.

***NO:** The text transparency feature is not used.

***YES:** The text transparency feature is used which permits the transmission of all 256 EBCDIC character codes. *YES is a valid value only if BLOCK(*NONE), BLOCK(*NOSEP), or BLOCK(*USER) is specified.

DTACPR
Specifies whether data compression is performed.

Note: DTACPR(*YES) cannot be specified if TRNSPY(*YES) or TRUNC(*YES) is specified. This parameter is valid only if APPTYPE(*BSCEL) or APPTYPE(*RPGT) is specified.

***SAME:** The value does not change.

***NO:** No data compression or decompression occurs.

***YES:** Data is compressed for output and decompressed for input.

TRUNC
Specifies whether trailing blanks are removed from output records. TRUNC(*YES) cannot be specified if BLOCK(*NOSEP), TRNSPY(*YES), or DTACPR(*YES) is specified. This parameter is valid only if APPTYPE(*BSCEL) or APPTYPE(*RPGT) is specified.

***SAME:** The value does not change.

***NO:** Trailing blanks are not removed from output records.

***YES:** Trailing blanks are removed from output records.

GRPSEP
Specifies a separator for groups of data, such as data sets and documents. This parameter is valid only if APPTYPE(*BSCEL) or APPTYPE(*RPGT) is specified.

***SAME:** The value does not change.

***EOT:** An end-of-transmission (EOT) control character is used.

***OFCSYS:** A transmission block ending with an end-of-text (ETX) control character is used.

***DEV3740:** A null record (STX ETX) is used.

EMLDEV
Specifies the type of 3270 device being emulated. This parameter is valid only if *EML was specified on the APPTYPE parameter when the device was created. The change requested by this command becomes effective the next time either the Start 3270 Display Emulation (STREML3270) command or the Start Print Emulation (STRPRTEML) command accesses the device.

***SAME:** The value does not change.

3278: This device description is used to emulate a 3278 display device.

3284: This device description is used to emulate a 3284 printer device.

3286: This device description is used to emulate a 3286 printer device.

3287: This device description is used to emulate a 3287 printer device.

3288: This device description is used to emulate a 3288 printer device.

3289: This device description is used to emulate a 3289 printer device.

EMLKBD

Specifies the type of 3278 display keyboard being emulated. A value is specified for this parameter only if this is a 3270 device emulation application type (APPTYPE is *EML). The specified value is used if EMLDEV(3278) was specified.

***SAME:** The value does not change.

***UPPER:** A 3270 display device keyboard is emulated with uppercase characters only.

***LOWER:** A 3270 display device keyboard is emulated with uppercase and lowercase characters.

EMLNUMLCK

Specifies whether numeric input fields allow only numeric data on a 5250 keyboard. The value can be specified for this parameter only if this is a 3270 device emulation application type (APPTYPE is *EML).

***SAME:** The value does not change.

***NO:** 3270 emulation allows any data to be typed in the numeric input fields.

***YES:** 3270 emulation allows only numeric data to be typed in the numeric input fields. Numeric data that can be typed include the characters 0 through 9, and symbols + - , . and blank.

EMLWRKSTN

Specifies the emulation work station that associates an emulation device with a real display or printer device. The device address is reserved for use exclusively by that work station. If no device or *ANY is specified, any work station can use the emulation device. A value can be specified for this parameter only if this is a 3270 device emulation application type (APPTYPE is *EML).

***SAME:** The value does not change.

***ANY:** Any work station can use the emulation device.

emulation-work-station: Specify the device name for the work station that uses this emulation device.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

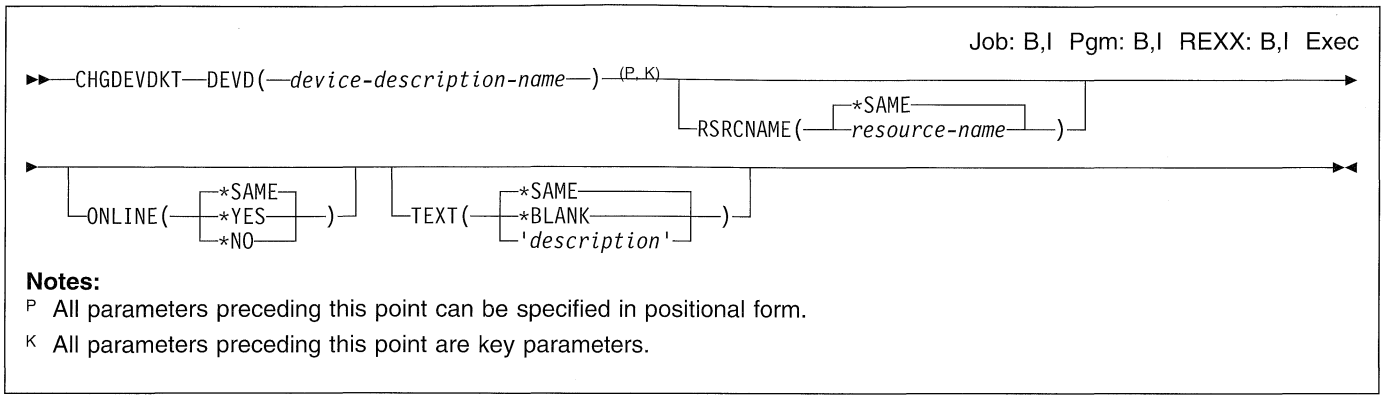
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGDEVBSC  DEVD(BSC001)  EMLDEV(3278)  EMLKBD(*LOWER)
```

This command changes the device description for the BSC device named BSC001 so it emulates a 3278 display station with both uppercase and lowercase characters.

CHGDEVDKT (Change Device Description (Diskette)) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Device Description (Diskette) (CHGDEVDKT) command changes the description of a diskette unit.

Required Parameter

DEVD

Specifies the name of the device description being changed.

Optional Parameters

RSRCNAME

Specifies the resource name that describes the automatic call unit port.

Note: Use the Work with Hardware Resources (WRKHDWRSC) command with *STG specified for the TYPE parameter to determine the resource name.

***SAME:** The value does not change.

resource-name: Specify the name that identifies the diskette unit on the system.

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The device is automatically varied on at IPL.

***NO:** This device is not automatically varied on at IPL.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

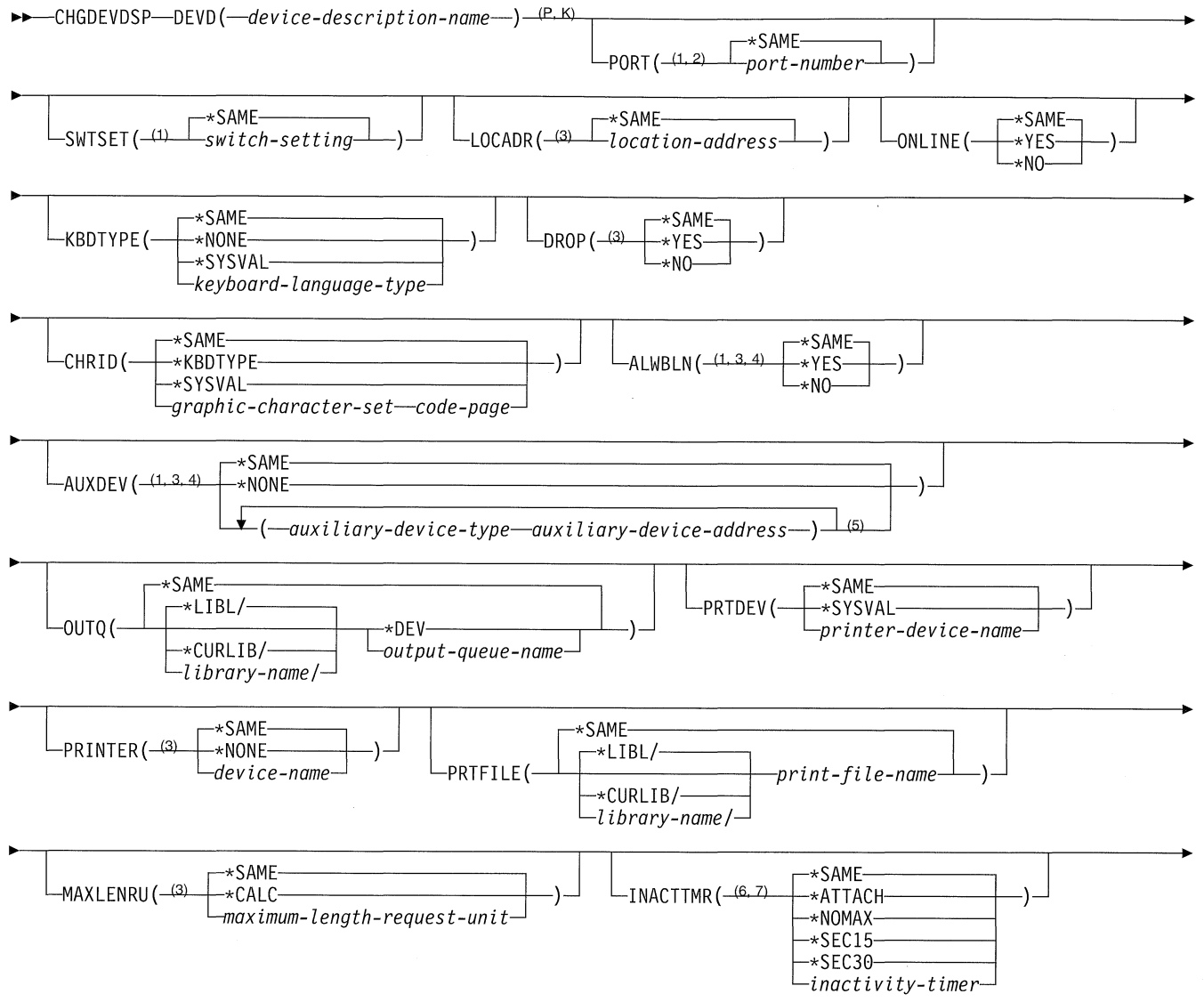
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGDEVDKT   DEVD(DKT01)  ONLINE(*YES)
```

This command changes the device description for DKT01 so the device description is varied on automatically during IPL.

CHGDEVDSP (Change Device Description (Display)) Command



Notes:

1 Local twinaxial devices.

2 Devices attached to ASCII work station controller, or SNPT or *NRF devices

3 Remote devices.

4 Virtual devices.

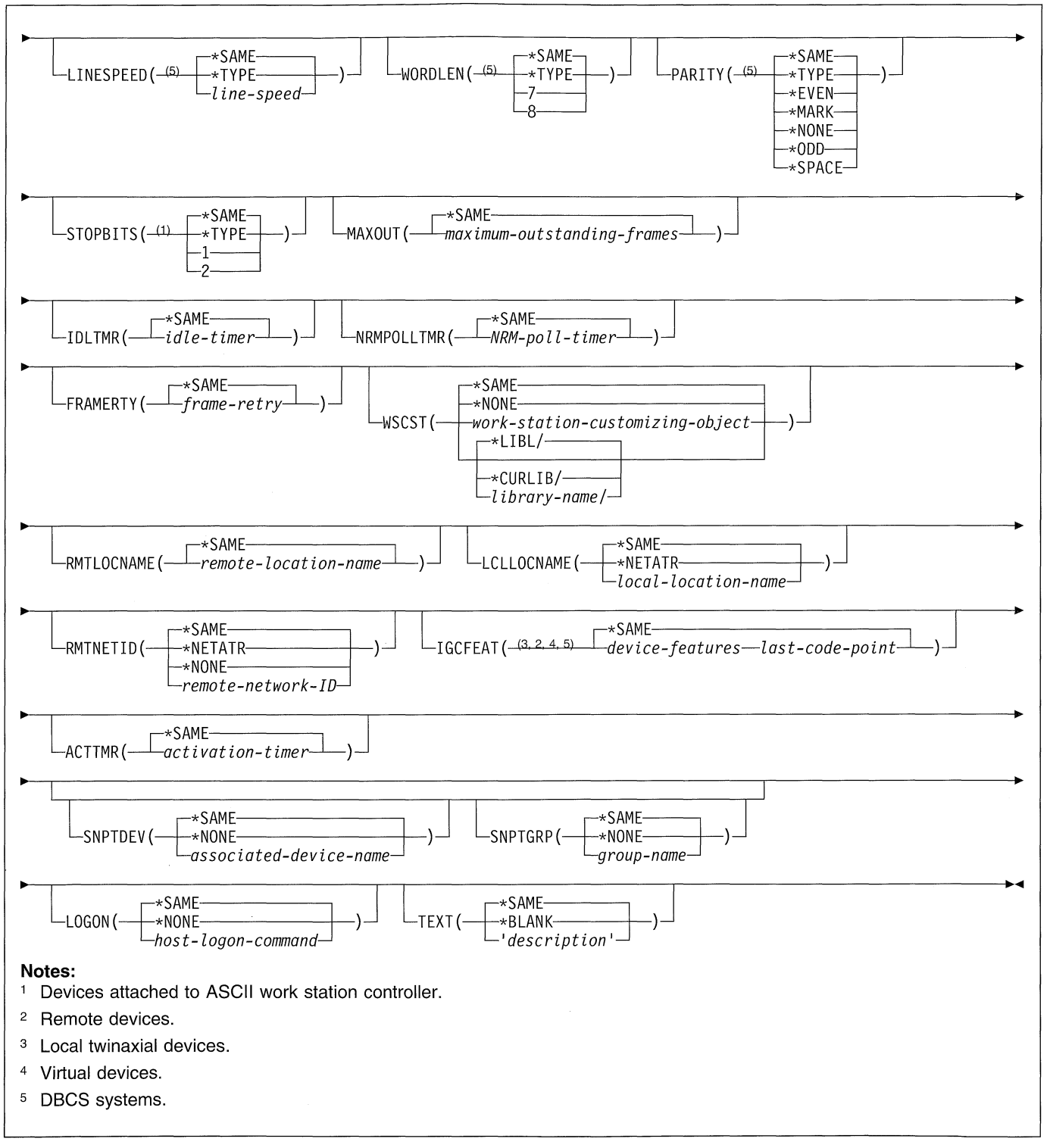
5 A maximum of 31 repetitions.

6 Devices attached to ASCII work station controller.

7 3270 devices.

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.



Notes:

1 Devices attached to ASCII work station controller.

2 Remote devices.

3 Local twinaxial devices.

4 Virtual devices.

5 DBCS systems.

Purpose

The Change Device Description (Display)) (CHGDEVDSP) command changes the description of a display device.

Restriction: This command cannot be used to remove port sharing (when *CALC was specified for the TYPE and/or LINESPEED parameters of the CRTDEVDSP command).

Port sharing for a display device must be removed by first deleting and then recreating its device description.

Required Parameter

DEVD

Specifies the name of the device description being changed.

Optional Parameters

PORT

Specifies, for local display stations, the port number for this display device. Valid values range from 0 through 17.

For devices attached to the ASCII work station controller only, valid values range from 0 through 17 and indicate the port of the ASCII work station controller to which this display station is attached. Without the 12-port expansion feature, only ports 0 through 5 are valid. With the 12-port expansion feature, ports 6 through 17 are added.

***SAME:** The value does not change.

port-number: Specify a value ranging from 0 through 17.

SWTSET

Specifies, for local twinaxial display stations, the switch setting or device address of the device. Does not apply to ASCII devices.

***SAME:** The value does not change.

switch-setting: Specify a value ranging from 0 through 6.

LOCADR

Specifies the local location address. Specify a 2-character hexadecimal value ranging from 00 through FE.

***SAME:** The value does not change.

location-address: Specify two hexadecimal characters with a value ranging from 00 through FE for the local location address. The value specified depends on the type of controller that is attached to the display.

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The device is automatically varied on at IPL.

***NO:** This device is not automatically varied on at IPL.

KBDTYPE

Specifies the country keyboard language identifier for this display station.

***SAME:** The value does not change.

***NONE:** No country keyboard language identifier is set for this display station. This is the only value for display stations for which a keyboard type is not required.

***SYSVAL:** The value specified in the system value QKBDTYPE is used.

keyboard-language-type: Specify the 3-character country identifier (used for EBCDIC and ASCII) for this display station.

Table 10 shows the valid identifiers and the language the identifier represents. In addition, Table 10 shows the ASCII device group(s), if applicable, for each language. The device group can be used to determine whether a language identifier can be used with an ASCII device. Valid display devices and the associated ASCII device group are shown in Table 11. These two tables can be used to determine whether an ASCII display device can be used with a specified language or if an ASCII device can be used at all.

For example, assume a user wants to create or change a 3101 display device. Table 11 shows that a 3101 display supports ASCII device group A. Table 10 shows the valid language identifiers that can be used with device group A include AGB, AGI, CAB, CAI, FAB, FAI, ITB, ITI, UKB, UKI, USB, and USI.

Table 10 (Page 1 of 2). Keyboard Mapping Table

Language/Country	Identifier	ASCII Device Group
Arabic X/Basic	CLB	D
Austria/Germany	AGB	A, B
Austria/Germany Multinational	AGI	A, B
Belgium Multinational	BLI	B
Brazilian Portuguese	BRB	
Canadian French	CAB	A, B
Canadian French Multinational	CAI	A, B
Cyrillic	CYB	
Denmark	DMB	B
Denmark Multinational	DMI	B
Finland/Sweden	FNB	B
Finland/Sweden Multinational	FNI	B
France (Azerty)	FAB	A, B
France (Azerty) Multinational	FAI	A, B
France (Qwerty)	FQB	
France (Qwerty) Multinational	FQI	
Greece	GNB	
Greece	GKB	
Hebrew	NCB	D
Iceland	ICB	
Iceland Multinational	ICI	
International	INB	
International Multinational	INI	
Italy	ITB	A, B
Italy Multinational	ITI	A, B
Japan English	JEB	
Japan English Multinational	JEI	
Japan Kanji (For PS*/55 and 5295 display stations)	JKB	
Japan United States Basic	JUB	
Japan Katakana (For 5251, 5291, 5292, and 3180 Katakana display stations)	KAB	
Korea	KOB	
Latin-2/ROECE	ROB	
Netherlands	NEB	
Netherlands Multinational	NEI	
Norway	NWB	B
Norway Multinational	NWI	B
Portugal	PRB	B
Portugal Multinational	PRI	B
Simplified Chinese	RCB	
Spain	SPB	B
Spain Multinational	SPI	B
Spanish Speaking	SSB	B

Table 10 (Page 2 of 2). Keyboard Mapping Table

Language/Country	Identifier	ASCII Device Group
Spanish Speaking Multinational	SSI	B
Sweden	SWB	B
Sweden Multinational	SWI	B
Switzerland/French Multinational	SFI	B
Switzerland/German Multinational	SGI	B
Thai	THB	
Traditional Chinese	TAB	
Turkey	TKB	
United Kingdom	UKB	A, B
United Kingdom Multinational	UKI	A, B
United States/Canada	USB	A, B, C
United States/Canada Multinational	USI	A, B, C
Languages of the former Yugoslavia	YGI	

Table 11 shows the ASCII device groups that are supported for each ASCII display. Refer to Table 10 to determine which languages can be used with the ASCII device group. Note the ASCII device group(s) to which each language belongs and, in Table 11, note the display stations that support each ASCII device group:

Table 11. Display/Device Group Table

ASCII Display		ASCII Device Group
3101		A
3151		B
3161		B
3162		B
3163		B
3164		B
D220	(Data General Dasher** D220)	C
T910	(TeleVideo** 910)	C
T925	(TeleVideo 925)	C
T955	(TeleVideo 955)	C
V100	(DEC** VT-100)	C
V220	(DEC VT-220)	C, D
W30	(Wyse** WY30**)	C
W50	(Wyse WY50**)	C
W60	(Wyse WY60**)	C

Notes:

1. The following values (*SYSVAL, AGI, BLI, CAI, DMI, FNI, FAI, ITI, NWI, PRI, SPI, SSI, SWI, USB, or USI) are also allowed when DEVCLS(*RMT) is specified and when TYPE(3277), TYPE(3278), or TYPE(3279) is specified. Otherwise, this parameter is not valid when DEVCLS(*RMT) or DEVCLS(*VRT) is specified.
2. This parameter is optional for the combination of DEVCLS(*LCL) and TYPE(5150).
3. This parameter is not valid for DEVCLS(*RMT) except as stated in the first note.

DROP

Specifies, for remote display stations, whether the line is disconnected (dropped) by the system when work stations on the line are no longer in use. When several work stations are attached to the same controller, the line is disconnected only if: (1) the device description for this device specifies DROP (*YES) or DROP (*YES) is specified on the SIGNOFF command when the user signs off the device; (2) all of the other display stations connected to the controller have signed off or are not in use; and (3) all printers attached to the controller are not in use.

The value specified in the device description is overridden by a user signing off at the device if the DROP parameter is used on the SIGNOFF command.

***SAME:** The value does not change.

***YES:** The switched line to the controller to which this device is attached is disconnected when this device and all other attached devices are no longer in use.

***NO:** The switched line to the controller is not disconnected when all of the attached devices are no longer in use.

CHRID

Specifies the character identifier (graphic character set and code page) that a work station display device supports. When a display file that was created with the CHRID DDS keyword is used with the device, the system converts data sent to and received from the device to ensure that the correct characters are shown and that the correct hexadecimal byte values are returned to the application program. More information about display file CHRID processing and the translation tables that are used to convert data sent to and received from the display are in the *Guide to Programming Displays*.

***SAME:** The value does not change.

***KBDTYPE:** The system determines the best graphic character set and code page for the keyboard specified on the KBDTYPE parameter.

***SYSVAL:** The system determines the graphic character set and code page values for the command parameters from the QCHRID system values.

Element 1: Character Set

graphic-character-set: Specify the graphic character set values that match the attributes of the display device. Valid values range from 1 through 32767.

Element 2: Code Page

code-page: Specify the code page set values that match the attributes of the display device. Valid values range from 1 through 32767.

The CHRID value specified is based on the attributes of the display device. Table 12 shows CHRID values that are appropriate for each work station display keyboard style. For some display devices the KBDTYPE param-

eter need not be specified, but the KBDTYPE value for the equivalent keyboard can be used to determine the CHRID value for the device.

All characters included in the graphic character sets listed for the *Limited CHRID* values can be both entered at and displayed by the device. All characters associated with the *Full CHRID* values can be entered at the display station (including the use of hexadecimal representations), but the device may not be able to display all the characters.

Values shown in the *Limited CHRID* column should be used for 5291 and 5292 display stations; all other display stations should use the value shown in the *Full CHRID* column. All display stations that are to be used for OfficeVision/400, including 5291 and 5292 display stations, should use the value in the *Full CHRID* column.

Note: When using keyboard type values (KBDTYPE parameter) for devices that support both single-byte and double-byte data (JKB, KOB, RCB, and TAB), the CHRID values listed are for single-byte data only.

ALWBLN

Specifies suppression of the (program controlled) blinking cursor. A blinking cursor may distract the operator.

***SAME:** The value does not change.

***YES:** Allows the cursor to blink for the 3179, 3180, 3196, 3197, 3476, 3477, 3486, 3487, 5251, 5252, 5291, and 5292 display devices.

***NO:** Cursor blinking is suppressed.

Note: For displays on which the blinking cursor attribute can be changed by using the keyboard setup feature, the value for ALWBLN in the device description may be overridden.

AUXDEV

Specifies the device type and address of an auxiliary device (if any) attached to the IEEE-488 port on the 5292 Model 2 device (or a PS/2* emulating a 5292 Model 2 graphics display). Up to 31 plotters may be attached to the same IEEE-488 AUXDEV port on the 5292 Model 2, but they must have different IEEE-488 addresses.

***SAME:** The value does not change.

***NONE:** No auxiliary device is specified.

Element 1: Auxiliary Device Types

auxiliary-device-type: Specify the auxiliary device (IBM plotter) that is attached to the 5292 Model 2 device. Valid plotters include the following:

7371	6180	6184	6186-1
7372	6182	6185	6186-2

Element 2: Auxiliary Device Addresses

auxiliary-device-address: Specify the address, ranging from 1 to 31, of the auxiliary device (plotter). Each plotter must have a unique address.

OUTQ

Specifies the qualified name of the output queue used by this user. The output queue must already exist.

***SAME:** The value does not change.

The name of the output queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

***DEV:** The output queue specified on the PRTDEV parameter is used.

output-queue-name: Specify the name of the output queue.

PRTDEV

Specifies the qualified name of the default printer device for this job. If OUTQ(*DEV) is specified, the file is placed on an output queue with the same name as the printer.

***SAME:** The value does not change.

***SYSVAL:** The value specified in the system value QPRTDEV is used.

printer-device-name: Specify the name of a printer that is used to print the output for this user.

PRINTER

Specifies, for a remote display station, the device name of the printer associated with the display device. The device description of the work station printer named in this parameter must already have been created and must currently exist on the system. Both the printer and display device must be attached to the same controller. The relationship created by this parameter is used when a related printer (PRINT keyword in DDS) is referred to in a device file used to access this work station.

***SAME:** The value does not change.

***NONE:** No printer is associated with this display station.

device-name: Specify the name of the printer associated with this display station. The printer and the display station must be attached to the same controller. When printing double-byte character set (DBCS) data, specify a DBCS printer (5553 or 5583).

PRTFILE

Specifies an alternative printer device file used for processing the Print key on this display station.

***SAME:** The value does not change.

The name of the device file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

print-file-name: Specify the name of the printer device file that performs Print key processing for this display station.

MAXLENRU

Specifies the maximum request unit (RU) length (in bytes) allowed.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

maximum-length-request-unit: Specify a value, to be used as the maximum length for incoming request units. Values 241 and 247 bytes are valid only for devices attached to an X.25 controller.

INACTTMR

Specifies an inactivity timer (time-out) value for display devices. This parameter also specifies what happens when the time-out value is exceeded, dependent on other attributes of the device:

- For display stations attached to an ASCII work station controller, the user's job is canceled when the display station is inactive (no data is sent or received) for a period of time that exceeds the time-out value. The display station is automatically varied off and on again, resulting in a new sign-on display.
- For display devices connected using SNA pass-through (SNPT) support, the user is informed by a message to QSYSOPR and the session is ended when the amount of time that the device is not bound to a host application exceeds the time-out value. The user must reestablish the connection and session.
- For display devices with an application type value of *APPINIT, *DEVINIT, or *NRF, the session is ended when the device is inactive (the file opened against the device is closed and no additional requests to open files are received for the device) for a period of time that exceeds the time-out value.

Note: This timer is not used by devices allocated to a subsystem (normal interactive use) because the subsystem always has a file open for the device. The timer is used by batch jobs that open and close files for the device.

For a Post Telephone and Telegraph (*PTT) attachment, the valid values are *SEC15, *SEC30, and from 1 through 10 minutes. The default of *ATTACH maps to *SEC30 (30 seconds).

For a connection using SNA pass-through (*SNPT) device class support, the default of *ATTACH maps to *NOMAX.

For a device with an application type value of *APPINIT, *CLTSSN, *DEVINIT, or *NRF, the default of *ATTACH maps to 1 minute.

For all other attachments, valid values range from 1 through 30 minutes. The default of *ATTACH has a value of *NOMAX.

***SAME:** The value does not change.

***ATTACH:** This value varies by the value specified on the physical attachment (ATTACH) parameter and certain values on the application type (APPTYPE) and device class (DEVCLS) parameters.

***NOMAX:** Maximum inactivity time is not tracked.

***SEC15:** A 15-second time-out period is used.

***SEC30:** A 30-second time-out period is used.

inactivity-timer: Specify a time-out value in minutes.

LINESPEED

Specifies, for display stations attached to the ASCII work station controller, the line speed (in bits per second) used to communicate over the interface (attachment) between the ASCII work station controller and the display station.

***SAME:** The value does not change.

***TYPE:** Instructs the system to use the suggested setting for this device type.

Valid line speed values are: 150, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, 19200, and 38400 bits per second.

For modem and Post Telephone and Telegraph (PTT) attachments (ATTACH(*MODEM) or ATTACH(*PTT)), the line speed specified should be the line speed selected for the modem.

The valid line speeds for each display are as follows:

Device	Lines Per Second
3101	150, 300, 600, 1200, 1800, 2400, 4800, 9600 (*TYPE)
3151	150, 300, 600, 1200, 1800, 2400, 3600, 4800, 9600, 19200 (*TYPE), 38400
3161	150, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, 19200 (*TYPE)

3162	300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, 19200 (*TYPE)
3163	150, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, 19200 (*TYPE)
3164	150, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, 19200 (*TYPE)
5150 Model A1	300, 600, 1200, 1800, 2400, 3600, 4800, 9600, 19200 (*TYPE)
D220 (Data General Dasher D220)**	150, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, 19200 (*TYPE)
T910 (TeleVideo 910)	150, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, 19200 (*TYPE)
T925 (TeleVideo 925)	150, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, 19200 (*TYPE)
T955 (TeleVideo 955)	150, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, 19200 (*TYPE), 38400
V100 (DEC VT-100)**	150, 300, 600, 1200, 1800, 2400, 3600, 4800, 9600, 19200 (*TYPE)
V220 (DEC VT-220)	150, 300, 600, 1200, 2400, 4800, 9600, 19200 (*TYPE)
W30 (Wyse WY30**)**	150, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200 (*TYPE), 38400
W50 (Wyse WY50)**	150, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200 (*TYPE), 38400
W60 (Wyse WY60)**	150, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200 (*TYPE), 38400

WORDLEN

Specifies, for display stations attached to the ASCII work station controller, the word length (bits per character) used to communicate with this display station. All display stations support both 7-bit and 8-bit word lengths except for the 3101 and the D220 which only supports 7. *TYPE specifies 8-bit word lengths for all display stations, except for the 3101 and the D220 which will be 7. For the 5150 Model A1, *TYPE specifies *NONE for 8-bit word lengths and *EVEN for 7-bit word lengths.

For modem and PTT attachments (ATTACH(*MODEM) or ATTACH(*PTT)), the word length specified is the word length selected for the modem.

***SAME:** The value does not change.

***TYPE:** The system uses the suggested setting for this display station.

7: 7-bit word lengths are used.

8: The 8-bit word length is used.

PARITY

Specifies, for ASCII display stations, the type of parity used to communicate with this display station. *TYPE specifies *EVEN for all display stations except for the D220 which will be *MARK.

For modem and PTT attachments (ATTACH(*MODEM) or ATTACH(*PTT)), the parity specified should be the parity selected for the modem.

***SAME:** The value does not change.

***TYPE:** The system uses the suggested setting for this display station.

***EVEN:** Even parity is used.

***MARK:** Mark parity is used.

***NONE:** No parity bit is used.

***ODD:** Odd parity is used.

***SPACE:** Space parity is used.

The valid parity types for each display station are:

Device	Parity Types
3101	*SPACE, *MARK, *ODD, *EVEN
3151	*NONE, *SPACE, *MARK, *ODD, *EVEN
3161	*NONE, *SPACE, *MARK, *ODD, *EVEN
3162	*NONE, *SPACE, *MARK, *ODD, *EVEN
3163	*NONE, *SPACE, *MARK, *ODD, *EVEN
3164	*NONE, *SPACE, *MARK, *ODD, *EVEN
5150 Model A1	*NONE, *ODD, *EVEN, *MARK
D220 (Data General D220)	*MARK
T910 (TeleVideo 910)	*NONE, *ODD, *EVEN,*TYPE
T925 (TeleVideo 925)	*NONE, *SPACE, *MARK, *ODD, *EVEN
T955 (TeleVideo 955)	*NONE, *SPACE, *MARK, *ODD, *EVEN
V100 (DEC VT-100)	*NONE, *ODD, *EVEN
V220 (DEC VT-220)	*NONE, *SPACE, *MARK, *ODD, *EVEN
W30 (Wyse WY30)	*NONE, *MARK, *ODD, *EVEN
W50 (Wyse WY50)	*NONE, *MARK, *ODD, *EVEN

W60 (Wyse WY60) *NONE, *MARK, *ODD, *EVEN

STOPBITS

Specifies, for display stations attached to the ASCII work station controller, the number of stop bits used to communicate with this display station. All display stations support both 1 and 2 stop bits except for the DEC VT-100 which only supports 2 stop bits. *TYPE specifies 1 stop bit for all display stations except DEC VT-100 (which is 2 stop bits). For modem and PTT attachments (ATTACH(*MODEM) or ATTACH(*PTT)), the stop bits specified are the stop bits selected for the modem.

***SAME:** The value does not change.

***TYPE:** The system uses the suggested setting for this display station.

1: One stop bit is used.

2: Two stop bits are used.

MAXOUT

Specifies the maximum number of frames sent sequentially to a remote system before the remote system (the 5150 work station) must respond. The maximum number of frames must range from 1 through 7.

This parameter is valid only if TYPE(5150) MODEL(A1) is specified.

***SAME:** The value does not change.

maximum-outstanding-frames: Specify a value ranging from 1 through 7 for the maximum number of frames.

IDLTMR

Specifies the time (in 0.1 second intervals) that the system waits for a response. If no response is received, error recovery procedures are started. This parameter is valid only if TYPE(5150) MODEL(A1) is specified.

***SAME:** The value does not change.

idle-timer: Specify a value ranging from 10 through 250 in 0.1 second intervals.

NRMPOLLTMR

Specifies the interval (in 0.1 second intervals) for polling this device when it is in normal response mode (NRM).

***SAME:** The value does not change.

NRM-poll-timer: Specify a value ranging from 2 through 100 in 0.1 second intervals.

This parameter is valid only if TYPE(5150) MODEL(A1) is specified.

FRAMERTY

Specifies the number of retries for an unanswered command frame or unacknowledged information frame. This parameter is valid only if TYPE(5150) MODEL(A1) is specified.

***SAME:** The value does not change.

frame-retry: Specify a value ranging from 5 through 64 for the number of frame retries.

WSCST

Specifies the qualified name of a work station customizing object.

***SAME:** The value does not change.

***NONE:** No work station customizing object is specified.

work-station-customizing-object: Specify the work station customizing object.

The name of the work station customizing object can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

Note: If a work station customizing object is specified for the WSCST parameter, all country keyboard identifiers are valid for ASCII devices except for the following: FQB, FQI, INB, INI, JEB, JEI, JKB, JUB, KAB, KOB, RCB, and TAB. See Table 28 for a list of the country keyboard identifiers.

RMTLOCNAME

Specifies the remote location name of the system with which this object communicates.

Note: This parameter is required for APPTYPE(*APPINIT) devices. The remote location name for APPTYPE(*APPINIT) devices is the VTAM/NCP (Virtual Telecommunications Access Method/Network Control Program) name of the physical device.

***SAME:** The value does not change.

LCLLOCNAME

Specifies the local location name. When this parameter is specified with APPTYPE(*CTLSSN) or APPTYPE(*APPINIT), the local location name is the name of the independent logical unit (LU) in the network control program (NCP).

***SAME:** The value does not change.

***NETATR:** The LCLLOCNAME value specified in the system network attributes is used.

local-location-name: Specify the local location name.

RMTNETID

Specifies the name of the remote network identifier (ID). This parameter can be specified for APPTYPE(*APPINIT) devices.

***SAME:** The value does not change.

***NETATR:** The RMTNETID value specified in the system network attributes is used.

***NONE:** No remote network identifier (ID) is used.

remote-network-ID: Specify the name of the remote network identifier.

IGCFEAT

Specifies which double-byte character set (DBCS) table is used by the device and language. This parameter is valid for DBCS-capable devices only.

***SAME:** The value does not change.

Element 1: Features of the DBCS-Capable Device

device-features: Specify the device features from Table 13.

Element 2: Last Code Point

last-code-point: Specify the last code point from Table 13.

ACTTMR

Specifies, for switched lines, the amount of time (in seconds) that the SNA pass-through support waits for the device to respond to the activation request from the host AS/400 system. If the device does not respond within this time, it is considered not available.

***SAME:** The value does not change.

activate-time: Specify a number ranging from 1 through 2550 indicating the number of seconds before the device is considered not available.

SNPTDEV

Specifies the name of the associated SNA pass-through device that is attached to a host or advanced program-to-program communications (APPC) controller.

***SAME:** The value does not change.

***NONE:** No name is specified.

associated-device-name: Specify the name of a device that is attached to a host or an APPC controller that is associated with this device.

SNPTGRP

Specifies the name configured for a group of host devices in a configuration list. This indicates that this device is associated with any one of the devices in that group which is available.

***SAME:** The value does not change.

***NONE:** No name is specified.

group-name: Specify the name configured for a group of host devices that must be associated with this device.

LOGON

Specifies the sign-on (logon) text. This parameter is allowed when DEVCLS(*SNPT) or APPTYPE(*NRF) is specified. APPTYPE(*NRF) specifies the logon string that is sent to the host system when a request is made to establish a session. DEVCLS(*SNPT) specifies the sign-on text that is sent to the host system after starting SNA pass-through support.

This parameter also specifies the logon string that is sent to the system service control point (SSCP) on the host network when the file is opened for *NRF.

***SAME:** The value does not change.

***NONE:** No text is sent to the host system.

host-logon-command: Specify text that is sent to the host system. The text must be enclosed in apostrophes if it contains blanks or other special characters. All apostrophes within the text must be represented by two apostrophes. A maximum of 256 characters can be specified.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Table 12 (Page 1 of 2). CHRID Values

Language/Country	Country Keyboard Language Identifier (KDBTYPE)	Limited CHRID	Full CHRID
International and US ASCII	INB	103 038	697 500
Multinational	AGI BLI CAI DMI FAI FNI FQI ICI INI ITI JEI NEI NWI PRI SFI SGI SPI SSI SWI UKI USI		697 500
Arabic	CLB		235 420
Austria/Germany	AGB	265 273	697 273
Belgium Multinational	BLI		697 500
Brazilian Portuguese	BRB		697 037
Canada/French	CAB	277 260	341 260
Cyrillic	CYB		960 880
Denmark/Norway	DMB NWB	281 277	697 277
Finland/Sweden	FNB SWB	285 278	697 278
France	FAB FQB	288 297	697 297

Table 12 (Page 2 of 2). CHRID Values

Language/Country	Country Keyboard Language Identifier (KDBTYPE)	Limited CHRID	Full CHRID
Greece	GNB GKB		925 875
Hebrew	NCB		941 424
Iceland	ICB		697 871
Italy	ITB	293 280	697 280
Japan/English	JEB	297 281	697 281
Japan/Kanji	JKB (For Personal System/55*, 5295 and 3477-J display stations)		1172 290
Japan/Katakana	KAB (For 5251, 5291, 5292, and 3180 Katakana display stations)		332 290
Korean	KOB		1173 833
Latin 2	ROB		959 870
Netherlands	NEB		697 037
Portugal	PRB	301 037	697 037
Simplified Chinese	RCB		1174 836
Spain	SPB	305 284	697 284
Spanish Speaking	SSB	309 284	697 284
Switzerland/French Multinational	SFI		697 500
Switzerland/German Multinational	SGI		697 500
Thai	THB		938 838
Traditional Chinese	TAB		101 037
Turkey	TKB		1152 1026
United Kingdom/English	UKB	313 285	697 285
United States/English	USB	101 037	697 037
Languages of the former Yugoslavia	YGI		959 870

Table 13. DBCS-Capable Device Features

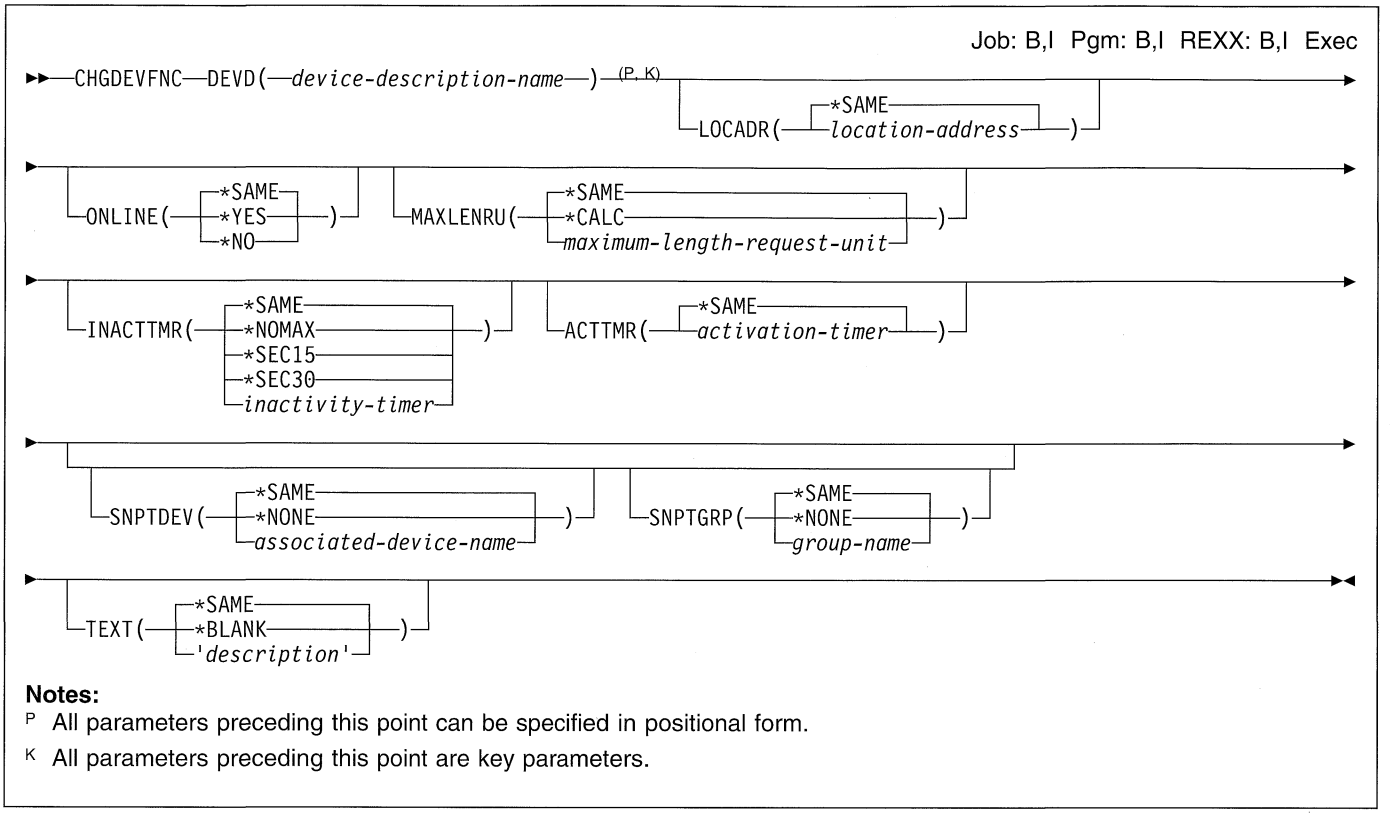
Language/ Device	Type of Physical DBCS-Capable Work Station	Configure as Type-Model	Configure with DBCS Feature
Japanese Display Stations	5295-001 Display Station	5555-B01	2424J4 55FE
	5295-002 Display Station	5555-B01	2424J4 68FE
	Personal System/55* running 5250PC	5555-B01	2424J4 68FE
	Personal System/55 running graphics-capable 5250PC	5555-G01	2424J4 68FE
	Personal System/55 running graphics-capable 5250PC	5555-G02	2424J4 68FE
	Personal System/55 running 5250PC/2	5555-E01	2424J0
Personal System/55 running PC Support/400 by OS/400 system	5555-B01	2424J0	
3270-type Display Stations*	3279-0	2424J0	
Korean Display Stations	5250-Type Display Stations	5555-B01	2424K0
	3270-type Display Stations*	3279-0	2424K0
Traditional Chinese Display Stations	5250-Type Display Stations	5555-B01	2424C0
	3270-type Display Stations*	3279-0	2424C0
Simplified Chinese Display Stations	5250-Type Display Stations	5555-B01	2424S0
	3270-type Display Stations*	3279-0	2424S0

Example

```
CHGDEVDSP  DEVD(DSP4)  PORT(1)  SWTSET(5)
```

This command moves display station DSP4 to port 1 and changes its address to 5. No other device can exist at this location on the same controller.

CHGDEVFNC (Change Device Description (Finance)) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Device Description (Finance) (CHGDEVFNC) command changes the description of a finance device.

Required Parameter

DEVD

Specifies the name of the device description being changed.

Optional Parameters

LOCADR

Specifies the local location address. Specify a 2-character hexadecimal value ranging from 01 through FF.

***SAME:** The value does not change.

location-address: Specify 2 hexadecimal characters with a value ranging from 01 to FF for the local location address.

Controllers and valid numbers of attachable devices are as follows:

3694	01-04
4701	01-FF
4702	01-FF
4730	01-03
4731	01-02
4732	01-02
4736	01-02
***FBSS**	01-FF

Note: The value 01 is only valid for ICF finance communications when (TYPE(*FNCICF) is specified on the Create Device Description (Finance) (CRTDEVFNC) command.

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The device is automatically varied on at IPL.

***NO:** This device is not automatically varied on at IPL.

MAXLENRU

Specifies the default maximum request unit (RU) length that can be sent or received by the local systems if the maximum size is not specified in the bind.

***SAME:** The value does not change.

***CALC:** The system determines the value to use.

maximum-length-request-unit: Specify a value ranging from 8 to 4096 bytes for the maximum size of the request unit.

INACTTMR

Specifies an inactivity timer (time-out) value. This parameter also specifies what happens when the time-out value is exceeded, dependent on other attributes of the device:

- For devices connected using SNA pass-through (SNPT) support, the user is informed by a message to QSYSOPR and the session is ended when the amount of time that the device is not bound to a host application exceeds the time-out value. The user must reestablish the connection and session.
- For all other attachments, valid values range from 1 through 30 minutes.

Note: This timer is not used by devices allocated to a subsystem (normal interactive use) because the subsystem always has a file open for the device. The timer is used by batch jobs that open and close files for the device.

***SAME:** The value does not change.

***NOMAX:** Maximum inactivity time is not tracked.

***SEC15:** A 15-second time-out period is used.

***SEC30:** A 30-second time-out period is used.

inactivity-timer: Specify a time-out value.

ACTTMR

Specifies, for switched lines, the amount of time that the SNA passthrough support waits for the device to respond to the activation request from the host AS/400 system. If the device does not respond within this time, it is considered not available.

***SAME:** The value does not change.

activation-time: Specify a number ranging from 1 through 2550 indicating the number of seconds before the device is considered not available.

SNPTDEV

Specifies the name of the associated SNA pass-through device that is attached to a host or advanced program-to-program communications (APPC) controller.

***SAME:** The value does not change.

***NONE:** No name is specified.

associated-device-name: Specify the name of a device that is attached to a host or an APPC controller that is associated with this device.

SNPTGRP

Specifies the name configured for a group of host devices in a configuration list (see CRTCFGL command). This indicates that this device is associated with any one of the devices in that group which is available.

This parameter is only allowed when *SNPT is specified for the DEVCLS parameter.

***SAME:** The value does not change.

***NONE:** No name is specified.

group-name: Specify the name configured for a group of host devices that must be associated with this device.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

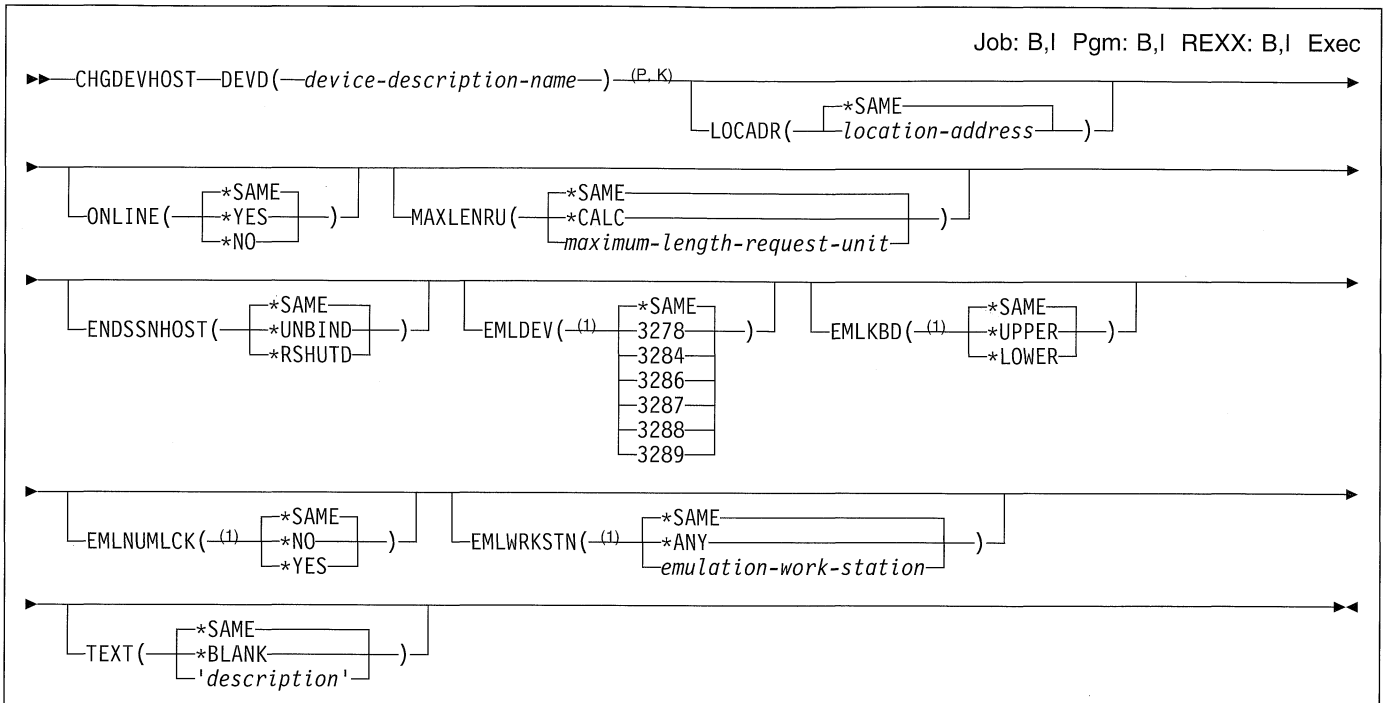
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGDEVFNC   DEVD(FNCDSP1)  ONLINE(*YES)
```

This command changes the device description for the finance device FNCDSP1 so that the device will be varied on at IPL.

CHGDEVHOST (Change Device Description (SNA Host)) Command



Notes:

1 This parameter is valid only when APPTYPE(*EML) is specified on the CRTDEVHOST command.

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Device Description (SNA Host) (CHGDEVHOST) command changes the description of a Systems Network Architecture (SNA) host system device.

Required Parameter

DEVD

Specifies the name of the device description being changed.

Optional Parameters

LOCADR

Specifies the local location address. Specify a 2-character hexadecimal value ranging from 01 through FF.

***SAME:** The value does not change.

location-address: Specify two hexadecimal characters with a permissible value ranging from 01 through FF for the local location address.

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The device is automatically varied on at IPL.

***NO:** This device is not automatically varied on at IPL.

MAXLENRU

Specifies the default maximum request unit (RU) length that can be sent or received by the local systems if the maximum size is not specified in the bind.

Note: This parameter is valid for devices attached to controllers of X.25 networks.

***SAME:** The value does not change.

***CALC:** The system determines the value to use.

maximum-length-request-unit: Specify a value, ranging from 256 through 4096 bytes, in 256-byte increments, to be used as the maximum length for incoming request units. Values of 241, 245, 247, 497, 501, and 503 bytes are valid only for devices attached to an X.25 controller.

ENDSSNHOST

Specifies how the host device ends a session with the host system.

Note: This parameter can be changed at any time and takes effect immediately after the change occurs.

***SAME:** The value does not change.

***UNBIND:** The host device sends the SNA command requesting the AS/400 system to end the session.

***RSHUTD:** The host device sends the SNA command requesting the host system to end the session.

EMLDEV

Specifies the type of 3270 device being emulated. This parameter is valid only if *EML was specified on the APPTYPE parameter when the device was created. The change requested by this command becomes effective the next time either the Start 3270 Display Emulation (STREML3270) command or the Start Print Emulation (STRPRTEML) command accesses the device.

***SAME:** The value does not change.

3278: This device description is used to emulate a 3278 display device.

3284: This device description is used to emulate a 3284 printer device.

3286: This device description is used to emulate a 3286 printer device.

3287: This device description is used to emulate a 3287 printer device.

3288: This device description is used to emulate a 3288 printer device.

3289: This device description is used to emulate a 3289 printer device.

EMLKBD

Specifies the type of 3278 display keyboard being emulated. A value is specified for this parameter only if APPTYPE(*EML) was specified when the device was created. The specified value is used if EMLDEV(3278) was specified.

***SAME:** The value does not change.

***UPPER:** A 3270 display device keyboard is emulated with uppercase characters only.

***LOWER:** A 3270 display device keyboard is emulated with both uppercase and lowercase characters.

EMLNUMLCK

Specifies whether numeric input fields will only allow numeric data on a 5250 keyboard. A value can be specified for this parameter only if this is a 3270 device emulation application type (APPTYPE is *EML).

***SAME:** The value does not change.

***NO:** 3270 emulation allows any data to be typed in the numeric input fields.

***YES:** 3270 emulation allows only numeric data to be typed in the numeric input fields. Numeric data that can be typed includes the characters 0 through 9 and the symbols + - , . and blank.

EMLWRKSTN

Specifies that this emulation device is associated with a real display device or printer device and that it is reserved for use exclusively by that work station. If no device or *ANY is specified, any work station can use the emulation device. A value can be specified for this parameter only if APPTYPE(*EML) is specified.

***SAME:** The value does not change.

***ANY:** Any work station can use the emulation device.

emulation-work-station: Specify the device name for the work station that uses this emulation device.

TEXT

Specifies text that briefly describes the device file. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

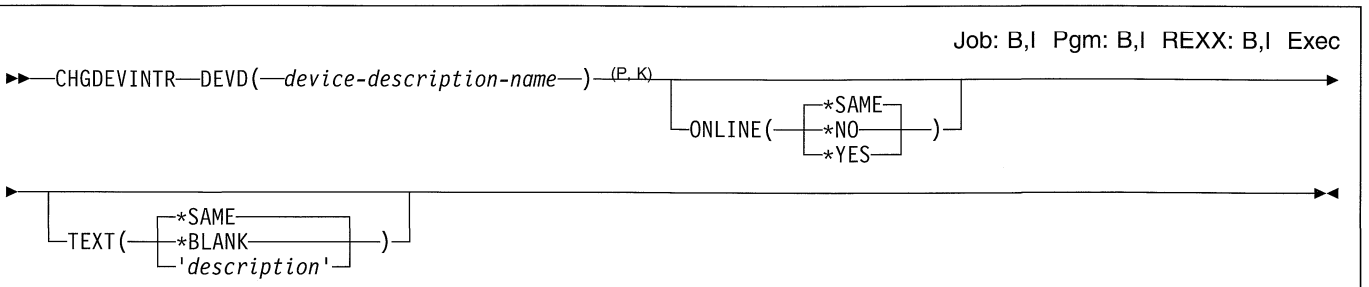
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGDEVHOST   DEVD(COMMDSP1)  EMLDEV(3286)
```

This command changes the device description named COMMDSP1 so it emulates a 3286 printer.

CHGDEVINTR (Change Device Description (Intrasystem)) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Device Description (Intrasystem) (CHGDEVINTR) command changes the description of an intrasystem device.

Required Parameter

DEVD

Specifies the name of the device description being changed.

Optional Parameters

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***NO:** This device is not automatically varied on at IPL.

***YES:** The device is automatically varied on at IPL.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

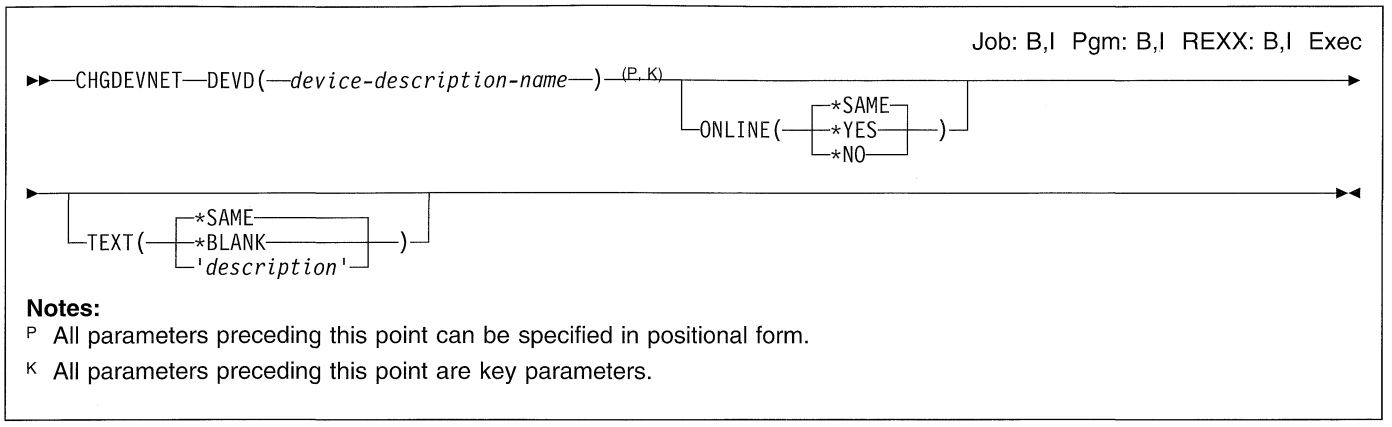
Example

```
CHGDEVINTR   DEVD(TRM5292)  ONLINE(*NO)
```

This command changes the ONLINE parameter to indicate that the device is not automatically varied on during IPL.

CHGDEVNET (Change Device Description (Network)) Command

command command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Device Description (Network) (CHGDEVNET) command changes the description of an open network device.

Required Parameter

DEVD

Specifies the name of the device description being changed.

Optional Parameters

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The device is automatically varied on at IPL.

***NO:** This device is not automatically varied on at IPL.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

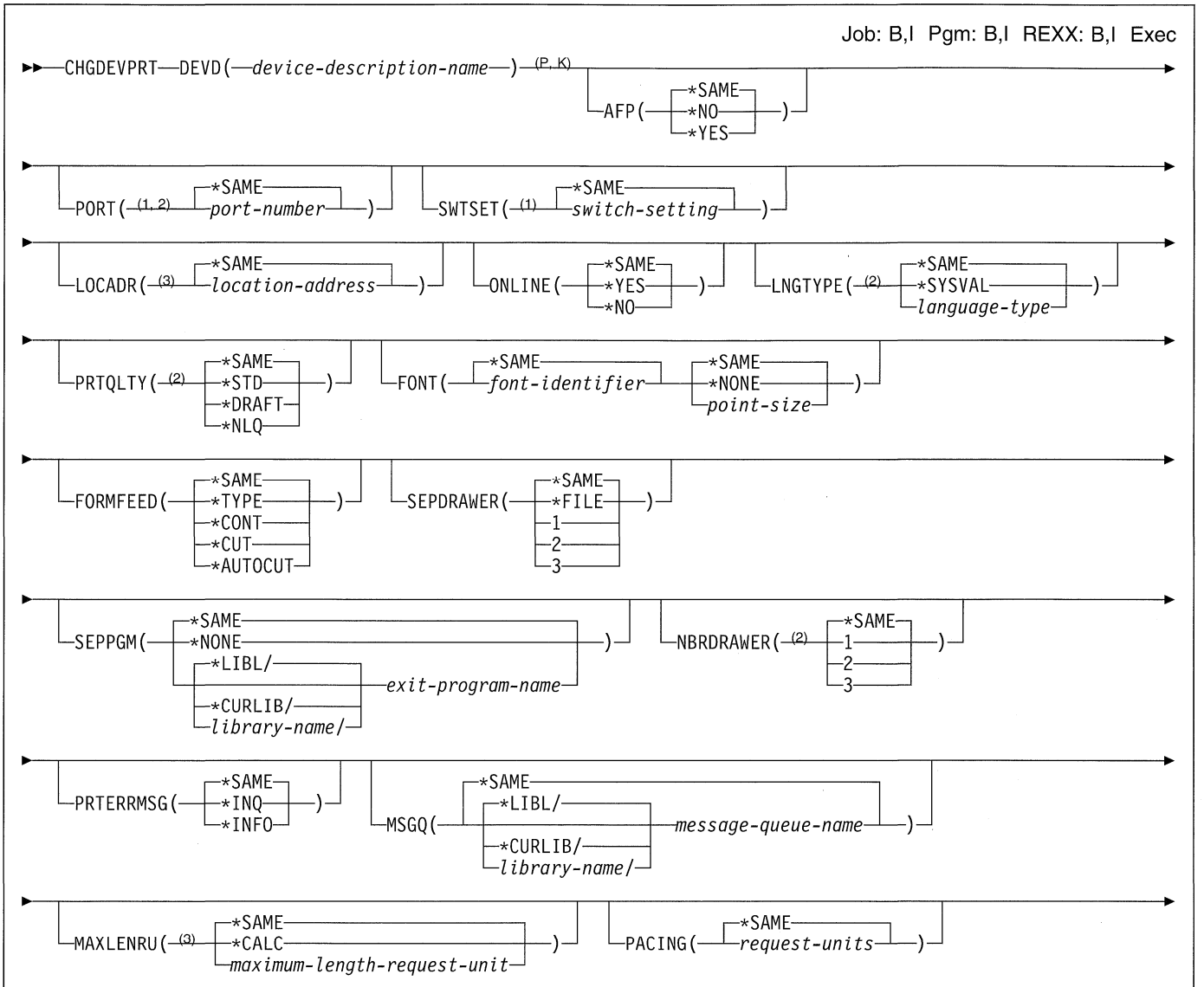
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGDEVNET   DEVD(NETDEV02)  ONLINE(*NO)
```

This command changes a device description for a network device named NETDEV02 so that the device is not automatically varied on at IPL.

CHGDEVPRT (Change Device Description (Printer)) Command



Notes:

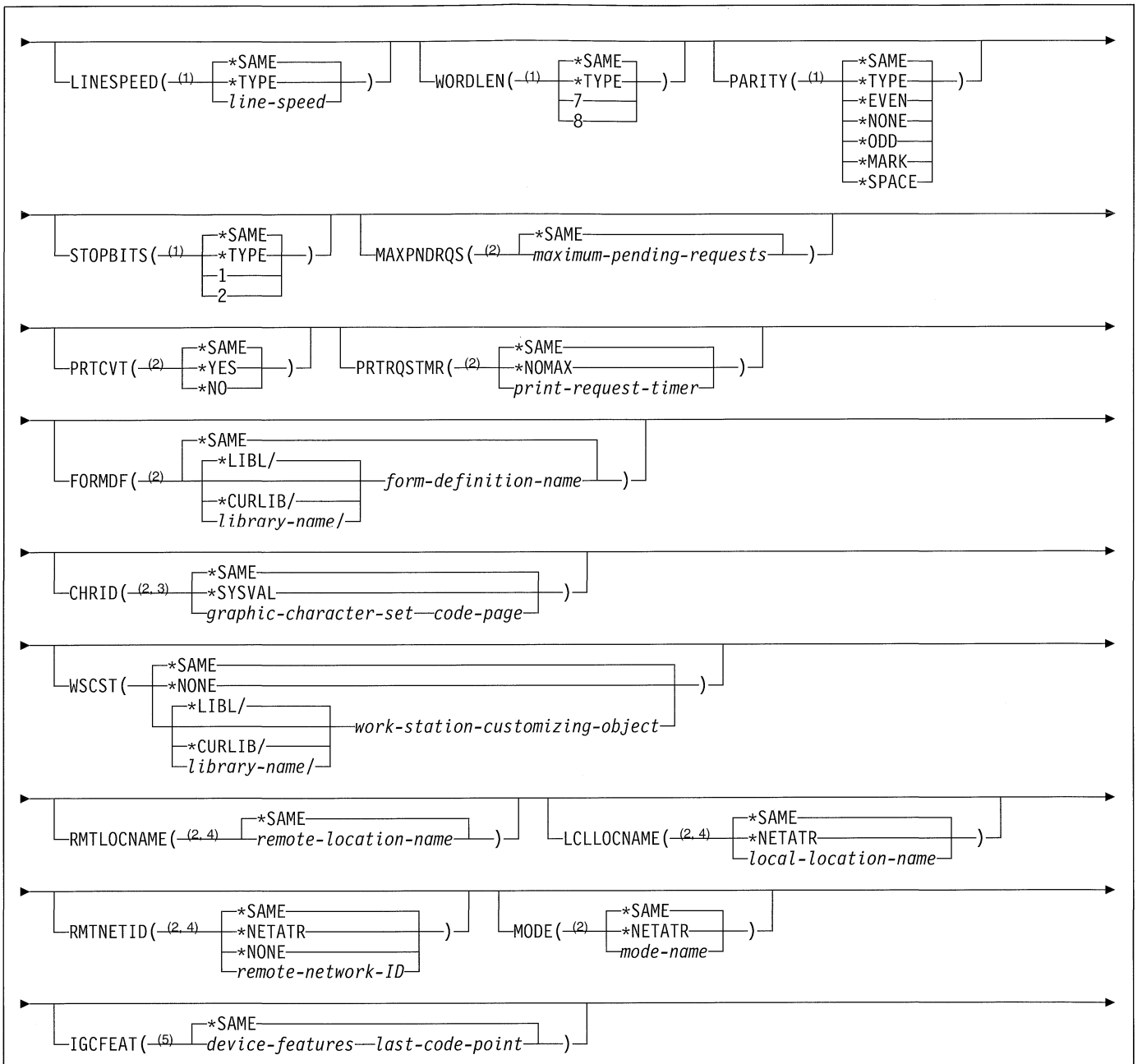
1 Local twinaxial devices.

2 Devices attached to an ASCII work station controller, or SNPT or *NRF devices.

3 Remote devices

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.



Notes:

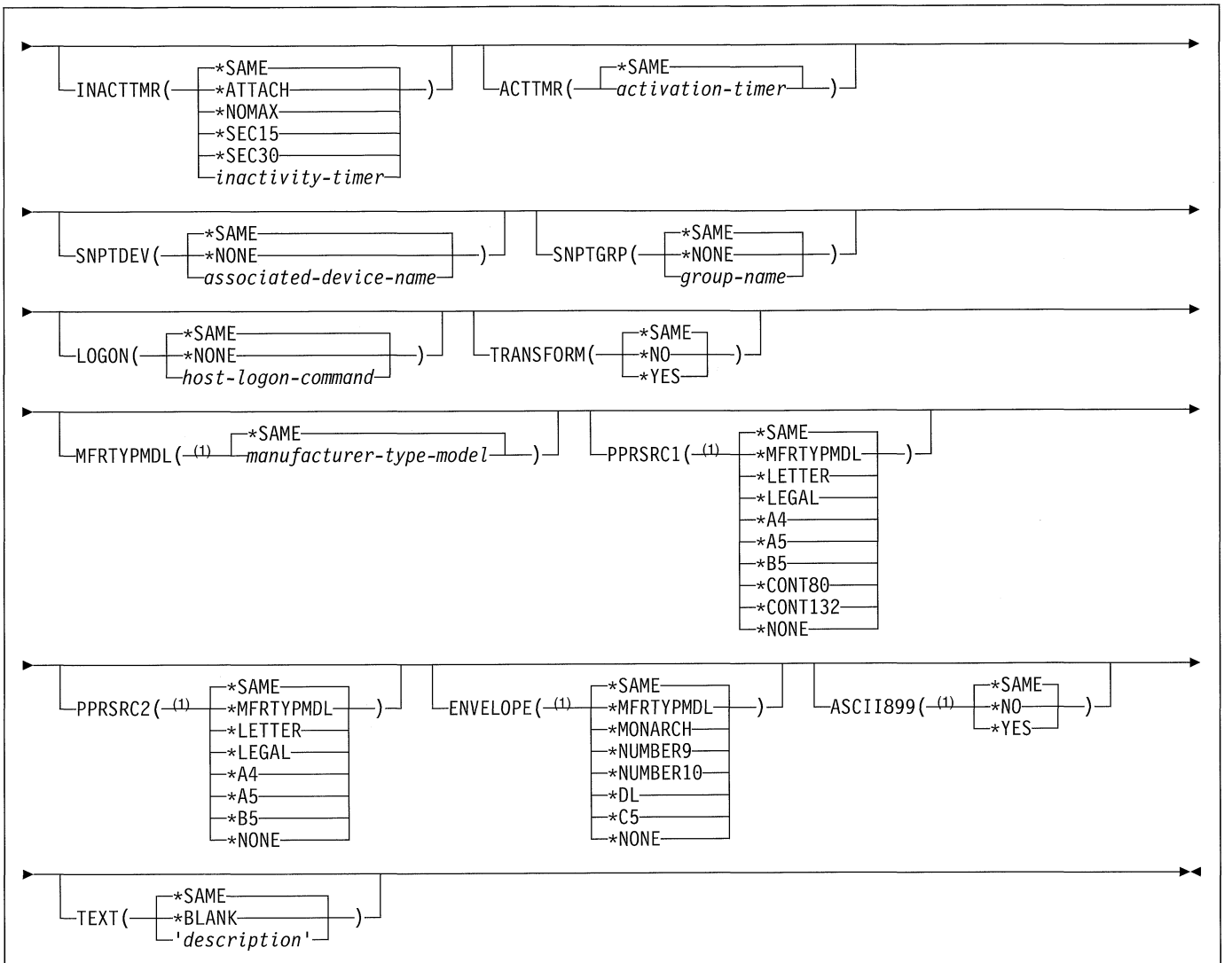
1 Devices attached to an ASCII work station controller, or SNPT or *NRF devices.

2 AFP printers

3 Devices using the host print transform.

4 Devices with Application type *APPINIT specified

5 DBCS systems



Note:
1 Devices using the host print transform.

Purpose

The Change Device Description (Printer) (CHGDEVPRT) command changes a device description for a printer device.

Required Parameter

DEVD

Specifies the name of the device description being changed.

Optional Parameters

AFP

Specifies whether the printer device does advanced function printing. This parameter is used only if DEVCLS (*LCL or *RMT) and TYPE(*IPDS) are specified.

Notes:

1. The DEVCLS and TYPE parameters are not changeable; they must be checked and their values specified while the user is specifying whether printers are to be configured for advanced function printing.
2. If you change the value on the AFP parameter from *NO to *YES, you must power down the printer to reset the power-on defaults (such as forms length).

Most IPDS printers can be configured with AFP(*YES) or AFP(*NO). Most non-IPDS printers can be configured only with AFP(*NO). If you change the value on this parameter from *NO to *YES when you run the CHGDEVPRT command, you must power down the printer to reset power on defaults.

***SAME:** The value does not change.

***NO:** The printer is not used for advanced function printing.

***YES:** The printer is used for advanced function printing.

PORT

Specifies the port number for the device. Valid values range from 0 through 17.

For ASCII devices only: Valid values range from 0 through 17 and indicate to which port of the ASCII work station controller the printer is attached. Without the 12-port expansion feature, only ports 0 through 5 are valid. With the 12-port expansion feature, ports 6 through 17 are added.

***SAME:** The value does not change.

port-number: Specify a value ranging from 0 through 17.

SWTSET

Specifies, for local twinaxial printers, the switch setting or device address for the device.

***SAME:** The value does not change.

switch-setting: Specify a value ranging from 0 through 6.

LOCADR

Specifies the local location address. Specify a 2-character hexadecimal value ranging from 00 through FE.

***SAME:** The value does not change.

location-address: Specify a 2-hexadecimal value ranging from 00 through 41.

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The device is automatically varied on at IPL.

***NO:** This device is not automatically varied on at IPL.

LNGTYPE

Specifies the default language (country code) for an ASCII printer.

***SAME:** The value does not change.

***SYSVAL:** The default language is *SYSVAL, which uses the QKBDTYPE system value.

language-type: Specify the language identifier code from the following table.

Table 14. Keyboard Mapping Table for Printers

Language/Country	Identifier	ASCII Device Group
Arabic X/Basic	CLB	D
Austria/Germany	AGB	A, B
Austria/Germany Multinational	AGI	A, B
Belgium Multinational	BLI	B
Brazilian Portuguese	BRB	
Canadian French	CAB	A, B
Canadian French Multinational	CAI	A, B
Cyrillic	CYB	
Denmark	DMB	B
Denmark Multinational	DMI	B
Finland/Sweden	FNB	B
Finland/Sweden Multinational	FNI	B
France (Azerty)	FAB	A, B
France (Azerty) Multinational	FAI	A, B
France (Qwerty)	FQB	
France (Qwerty) Multinational	FQI	
Greece	GKB	
Greece	GNB	
Hebrew	NCB	D
Iceland	ICB	
Iceland Multinational	ICI	
International	INB	
International Multinational	INI	
Italy	ITB	A, B
Italy Multinational	ITI	A, B
Japan English	JEB	
Japan English Multinational	JEI	
Japan Kanji (For PS/55* and 5295 display stations)	JKB	
Japan United States Basic	JUB	
Japan Katakana (For 5251, 5291, 5292, and 3180 Katakana display stations)	KAB	
Korea	KOB	
Latin-2/ROECE	ROB	
Netherlands	NEB	
Netherlands Multinational	NEI	
Norway	NWB	B
Norway Multinational	NWI	B
Portugal	PRB	B
Portugal Multinational	PRI	B
Simplified Chinese	RCB	
Spain	SPB	B
Spain Multinational	SPI	B
Spanish Speaking	SSB	B
Spanish Speaking Multinational	SSI	B
Sweden	SWB	B
Sweden Multinational	SWI	B
Switzerland/French Multinational	SFI	B
Switzerland/German Multinational	SGI	B
Thailand	THB	
Traditional Chinese	TAB	
Turkey	TKB	
United Kingdom	UKB	A, B
United Kingdom Multinational	UKI	A, B
United States/Canada	USB	A, B, C
United States/Canada Multinational	USI	A, B, C
Languages of the former Yugoslavia	YGI	

PRTQLTY

Specifies whether the default print quality for ASCII printers should be draft (*DRAFT), standard (*STD), or near-letter quality (*NLQ), from least to best quality. All

ASCII printer types (with all emulations) support this parameter. If the printer is emulating a 5219 (EMLDEV parameter), this quality setting is overridden by individual printer files sent to this printer.

***SAME:** The value does not change.

***STD:** The output is printed with standard quality.

***DRAFT:** Draft print quality setting is used.

***NLQ:** Near-letter print quality setting is used.

The Print Quality (PRTQLTY Parameter) table in the CRTDEVPRT description command shows the valid values for the 4214, 4224, 4234, and 5219 Printers.

FONT

Specifies the font identifier used by the 3812, 3816, 3820, 3825, 3827, 3829, 3835, 3900, and 5219 (including ASCII printers emulating the 5219 Printer), and IPDS printers.

The Printer Font Table (Font Parameter) in the CRTDEVPRT command lists the valid font identifiers, the display value, the characters per inch value implied with each font style, a description of each font style, and whether the font is supported on a particular printer.

Note: Some fonts may be substituted by the printer. Consult the various printer reference guides for details.

Element 1: Font Identifier

***SAME:** The value does not change.

font-identifier: Specify the 3-, 4-, or 5-digit font identifier associated with this printer.

Element 2: Point Size

***SAME:** The value does not change.

***NONE:** The point size is supplied by the system and is determined by the specified font identifier.

point-size: Specify a point size ranging from 0.1 through 999.9.

FORMFEED

Specifies, for the 4214, 5219, and 5553 printers (including ASCII printers that are configured as an SCS 4214 or SCS 5219 printer), and for IPDS printers, the mode in which forms are fed into the device.

Note: The FORMFEED parameter is overridden by the value specified on the PPRSRC1 parameter when the host print transform function is enabled.

***SAME:** The value does not change.

***TYPE:** Form feed value is chosen by the system based on the printer type.

***CONT:** Continuous forms are used by the printer. For ASCII devices only: Continuous forms are used by the printer; valid for all ASCII printers, except the 4216 Printer emulating a 5219 Printer.

***CUT:** Single-cut sheets are manually fed into the printer (for ASCII devices only). Valid for all ASCII printers only when the printers are emulating a 5219 Printer.

***AUTOCUT:** The sheet-feed attachment must be on the printer. Single-cut sheets are automatically fed into the printer. The forms alignment message is not sent for cut sheets. For ASCII devices only: Single-cut sheets are automatically fed into the printer; valid for 4207, 4208, 4216, 4224, and 5204 Printers that are emulating a 5219 Printer.

SEPDRAWER

Specifies which drawer is selected for printing separators.

***SAME:** The drawer specified for separator pages does not change.

***FILE:** The separator pages are printed on paper from the same drawer as the rest of the spooled file.

1: The separator pages are printed from drawer 1.

2: The separator pages are printed from drawer 2.

3: The separator pages are printed from drawer 3.

Note: SEPDRAWER(3) implies 2 drawers plus an envelope drawer.

SEPPGM

Specifies a style of separator page by allowing you to call a user exit program while printing job and file separators.

***SAME:** The value does not change.

***NONE:** The separator pages are not changed.

The name of the exit program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

exit-program-name: Specify an exit program name.

NBRDRAWER

Specifies, for ASCII devices only, when FORMFEED(*AUTOCUT) is specified, whether one drawer, two drawers, or three drawers are supported by the printer.

***SAME:** The value does not change.

1: One drawer is supported.

2: Two drawers are supported.

3: Three drawers are supported.

This parameter is not allowed with FORMFEED(*CONT) and FORMFEED(*CUT). For the printers that support FORMFEED(*AUTOCUT), the following numbers of drawers are supported:

- For 4207-1 and 4208 Printers, NBRDRAWER(1) is the only valid response.
- For a 4224 Printer, NBRDRAWER(3) is the only valid response.
- For a 4207-2, 4216, or 5204 Printer, NBRDRAWER values of 1, 2, or 3 drawers are supported.

Note: NBRDRAWER(3) implies 2 drawers plus an envelope drawer.

See the description of the NBRDRAWER parameter for the CRTDEVPRT command.

PRTERRMSG

Specifies whether the device sends inquiry messages or informational messages when recoverable printer errors occur.

***SAME:** The value does not change.

***INQ:** Inquiry messages for recoverable printer errors are sent whenever possible.

***INFO:** Informational messages are sent when recoverable printer errors occur.

MSGQ

Specifies the qualified name of the message queue to which messages are sent.

***SAME:** The value does not change.

The name of the message queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

message-queue-name: Specify the name of the message queue to which messages are sent.

MAXLENRU

Specifies the maximum request unit (RU) length (in bytes) allowed.

***SAME:** The value does not change.

***CALC:** The system determines the value to use.

maximum-length-request-unit: Specify a value for the maximum length for incoming request units. Values 241 and 247 are valid only for X.25 networks.

PACING

Specifies the rate at which the receiver can accept data (measured in request units) from the SNA data flow. This parameter cannot be specified for 3287 remote printers.

***SAME:** The value does not change.

request-units: Specify the number of request units that can be accepted by the receiver. Valid values range from 1 through 7.

LINESPEED

Specifies the line speed (in bits per second) that ASCII printers use to communicate over the interface (attachment) between the ASCII work station and the printer. The value of *TYPE specifies 19200 bits per second for all printers.

***SAME:** The value does not change.

***TYPE:** The system uses the suggested setting (in bits per second) for the device type. *TYPE specifies 19200 for all printers.

Valid values are 150, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, 19200, and 38400.

The valid line speeds for each printer are as follows:

Printer	Line Speed
4201-2	150, 300, 600, 1200, 2400, 4800, 9600, 19200
4201-3	150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400
4202-1	150, 300, 600, 1200, 2400, 4800, 9600, 19200
4202-2	150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400
4202-3	150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400
4207-1	150, 300, 600, 1200, 2400, 4800, 9600, 19200
4207-2	150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400
4208-1	150, 300, 600, 1200, 2400, 4800, 9600, 19200
4208-2	150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400
4216	600, 1200, 2400, 4800, 9600, 19200
4224	300, 600, 1200, 2400, 4800, 9600, 19200
4234	150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400
5204	300, 600, 1200, 2400, 4800, 9600, 19200
6252	300, 600, 1200, 2400, 9600, 19200, 38400

WORDLEN

Specifies the word length (bits per character) ASCII printers use to communicate with the printer.

Valid values are *TYPE, 7 bits, and 8 bits. All ASCII printers support both 7- and 8-bit word lengths except the 4224 Printer and the 6252 Printer which support only 8-bit word lengths. *TYPE specifies 8-bit word lengths for all printers.

***SAME:** The value does not change.

***TYPE:** The system uses the suggested setting for this printer.

7: 7-bit word lengths are used.

8: The 8-bit word length is used.

PARITY

Specifies the type of parity ASCII printers use to communicate with the printer. All printers support *NONE, *ODD, and *EVEN. *TYPE specifies *EVEN parity for all printers.

***SAME:** The value does not change.

***TYPE:** The system uses the suggested setting for this printer.

***EVEN:** Even parity is used.

***NONE:** No parity bit is used.

***ODD:** Odd parity is used.

***MARK:** Mark parity is used.

***SPACE:** Space parity is used.

STOPBITS

Specifies the number of stop bits ASCII printers use to communicate with the printer. All ASCII Printers support the use of both 1 and 2 stop bits, except the 4224, which only supports 1. *TYPE specifies the use of 1 stop bit for all printers.

***SAME:** The value does not change.

***TYPE:** The system uses the suggested setting for this printer.

1: One stop bit is used.

2: Two stop bits are used.

MAXPNDRQS

Specifies the maximum number of print requests that are queued for Advanced Function Printing* (AFP*) printers. This parameter is used only if AFP is *YES.

***SAME:** The value does not change.

maximum-pending-requests: Specify a number ranging from 1 to 31, indicating the maximum number of print requests that can be queued.

PRTCVT

Specifies whether a file (when using AFP) must be completely converted to IPDS before it can begin printing.

***SAME:** The value does not change.

***YES:** Printing begins prior to complete IPDS conversion.

***NO:** Printing does not begin prior to complete IPDS conversion.

PRTRQSTMR

Specifies the number of seconds to wait after a print request has been sent to a continuous forms printer configured for advanced function printing before the last printed output is forced into the stacker. This parameter is used only if AFP(*YES) and FORMFEED(*CONT) are specified.

***SAME:** The value does not change.

***NOMAX:** The print request timer is not used.

print-request-timer: Specify the time (seconds), ranging from 1 to 3600, for the print request timer.

FORMDF

Specifies the library and form definition name used for print requests that do not specify a form definition. This parameter is used only if AFP(*YES) is specified.

***SAME:** The value does not change.

The name of the form definition can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

form-definition-name: Specify the form definition name to use.

CHRID

Specifies the default character identifier (graphic character set and code page). This parameter is valid only if AFP(*YES) and AFPATTACH(*APPC) are specified, or if TRANSFORM(*YES) is specified.

Note: The AFPATTACH parameter is not changeable. To change the CHRID values for a printer, the printer must have been created by specifying AFPATTACH(*APPC) or TRANSFORM(*YES).

***SAME:** The value does not change.

***SYSVAL:** The system determines the graphic character set and code page values for the command parameters from the QCHRID system values.

Element 1: Character Set

graphic-character-set: Specify the graphic character set values that match the attributes of the printer. Valid values range from 1 through 32767.

Element 2: Code Page

code-page: Specify the code page set values that match the attributes of the printer. Valid values range from 1 through 32767.

WSCST

Specifies the qualified name of a work station customizing object.

***SAME:** The value does not change.

***NONE:** No work station customizing object is specified.

The name of the work station customizing object can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

work-station-customizing-object: Specify the work station customizing object.

Note: If a work station customizing object is specified for the WSCST parameter, all language keyboard country identifiers are valid for ASCII devices except for the following: FQB, FQI, INB, INI, JEB, JEI, JKB, JUB, KAB, KOB, RCB, and TAB. See Table 10 for a list of the language keyboard country identifiers.

RMTLOCNAME

Specifies the name of the remote system. This parameter is valid if AFP(*YES) and AFPATTACH(*APPC) are specified. This parameter is required when APPTYPE(*APPINIT) is specified. The remote location name for an APPTYPE(*APPINIT) device is the VTAM/NCP (Virtual Telecommunications Access Method/IBM Network Control Program) name of the physical device.

***SAME:** The value does not change.

remote-location-name: Specify the full name of a remote location.

LCLLOCNAME

Specifies the local location name. This parameter is valid only when AFP(*YES) and AFPATTACH(*APPC) are specified, or when APPTYPE(*APPINIT) is specified. The local location name for an APPTYPE(*APPINIT) device is the name of the independent logical unit (LU) in the network control program (NCP).

***SAME:** The value does not change.

***NETATR:** The LCLLOCNAME value specified in the system network attributes is used.

local-location-name: Specify the local location name.

RMTNETID

Specifies the identifier (ID) of the remote network. This parameter is required when AFP(*YES) and AFPATTACH(*APPC) is specified, or when APPTYPE(*APPINIT) is specified.

***SAME:** The value does not change.

***NETATR:** The RMTNETID value specified in the system network attributes is used.

***NONE:** No remote network identifier (ID) is used.

remote-network-ID: Specify the ID of the remote network.

MODE

Specifies the mode name used to define the communications session limits and session characteristics for the device. This parameter is valid only if AFP(*YES) and AFPATTACH(*APPC) are specified.

***SAME:** The value does not change.

***NETATR:** The mode name specified in the network attributes is used.

mode-name: Specify the mode name for the communications session.

IGCFEAT

Specifies which double-byte character set (DBCS) table is used by the device and language. This parameter is valid for DBCS-capable devices only.

***SAME:** The value does not change.

Element 1: Features of the DBCS-Capable Device

device-features: Specify the device features of the DBCS-capable device, which are in the DBCS-Capable Device Features (IGCFEAT Parameter) table located in the CRTDEVPRT command.

Element 2: Last Code Point

last-code-point: Specify the last code point of the DBCS-capable device, which is found in the DBCS-Capable Device Features (IGCFEAT Parameter) table located in the CRTDEVPRT command.

INACTTMR

Specifies an inactivity timer (time-out) value. This parameter also specifies what happens when the time-out value is exceeded, dependent on other attributes of the device:

- For devices connected using SNA pass-through (SNPT) support, the user is informed by a message to QSYSOPR and the session is ended when the amount of time that the device is not bound to a host application exceeds the time-out value. The user must reestablish the connection and session.
- For devices with an application type value of *APPINIT, *DEVINIT, or *NRF, the session is ended when the device is inactive (the file opened against the device is closed and no additional requests to open files are received for the device) for a period of time that exceeds the time-out value.

Note: This timer is not used by devices allocated to a subsystem (normal interactive use) because the subsystem always has a file open for the device. The timer is used by batch jobs that open and close files for the device.

For all other attachments, valid values range from 1 through 30 minutes.

***SAME:** The value does not change.

***ATTACH:** This value varies by the value on the physical attachment (ATTACH parameter) and certain values

on the device class (DEVCLS) and application type (APPTYPE) parameters. For DEVCLS(*SNPT) or APPTYPE(*DEVINIT) support, *ATTACH maps to *NOMAX. For APPTYPE(*NRF) and APPTYPE(*APPINIT) support, *ATTACH maps to 1 minute.

***NOMAX:** Maximum inactivity time is not tracked.

***SEC15:** A 15-second time-out period is used.

***SEC30:** A 30-second time-out period is used.

inactivity-timer: Specify a time-out value.

ACTTMR

Specifies, for switched lines, the amount of time (in seconds) that the SNA pass-through support waits for the device to respond to the activation request from the host AS/400 system. If the device does not respond within this time, it is considered not available.

***SAME:** The value does not change.

activation-timer: Specify a number ranging from 1 through 2550 indicating the number of seconds before the device is considered not available.

SNPTDEV

Specifies the name of the associated SNA pass-through device that is attached to a host or advanced program-to-program communications (APPC) controller.

***SAME:** The value does not change.

***NONE:** No name is specified.

associated-device-name: Specify the name of a device that is attached to a host or an APPC controller that is associated with this device.

SNPTGRP

Specifies the name configured for a group of host devices in a configuration list. This indicates that this device is associated with any one of the devices in that group which is available.

***SAME:** The value does not change.

***NONE:** No name is specified.

group-name: Specify the name configured for a group of host devices that must be associated with this device.

LOGON

Specifies the sign-on (logon) text. This parameter is allowed when DEVCLS(*SNPT) or APPTYPE(*NRF) is specified. APPTYPE(*NRF) specifies the logon string that is sent to the host system when a request is made to establish a session. DEVCLS(*SNPT) specifies the sign-on text that is sent to the host system after starting SNA pass-through support.

Specifies the logon string that is sent to the system service control point (SSCP) on the host network when the file is opened.

***SAME:** The value does not change.

***NONE:** No text is sent to the host system.

host-logon-command: Specify text that is sent to the host system. The text must be enclosed in apostrophes if it contains blanks or other special characters. All apostrophes within the text must be represented by two apostrophes. A maximum of 256 characters can be specified.

TRANSFORM

Specifies whether the printer uses the host print transform function.

***SAME:** The value does not change.

***NO:** The printer does not use the host print transform function.

***YES:** The printer uses the host print transform function.

Note: If TRANSFORM is changed from *NO to *YES, and host print transform has never been enabled for the device, the MFRTYPMDL parameter MUST be specified.

MFRTYPMDL

Specifies the manufacturer, type, and model for a printer using the host print transform function. See Table 33 on the CRTDEVPRT command for a list of the manufacturers, types, and models for printers using the host print transform function.

Note: If *WSCST is specified for MFRTYPMDL, a workstation customizing object must be specified.

***SAME:** The value does not change.

manufacturer-type-model: Specify a corresponding manufacturer, type, and model for a printer.

PPRSRC1

Specifies, for host print transform enabled printers, the type of paper in paper source one.

***SAME:** The value does not change.

***MFRTYPMDL:** The system uses the suggested setting for this printer.

***LETTER:** The paper for this source is letter-sized (8.5 x 11 inches).

***LEGAL:** The paper for this source is legal-sized (8.5 x 14 inches).

***EXECUTIVE:** The paper for this source is executive-sized (7.25 x 10.5 inches).

***A4:** The paper for this source is A4-sized (210mm x 297mm).

***A5:** The paper for this source is A5-sized (148 x 210mm).

***B5:** The paper for this source is B5-sized (182 x 257mm).

***CONT80:** The paper for this source is continuous form (8.0 inches).

***CONT132:** The paper for this source is continuous form (13.2 inches).

***NONE:** No paper source number one is specified.

PPRSRC2
Specifies, for host print transform enabled printers, the type of paper in paper source two.

***SAME:** The value does not change.

***MFRTYPMDL:** The system uses the suggested setting for this printer.

***LETTER:** The paper for this source is letter-sized (8.5 x 11 inches).

***LEGAL:** The paper for this source is legal-sized (8.5 x 14 inches).

***EXECUTIVE:** The paper for this source is executive-sized (7.25 x 10.5 inches).

***A4:** The paper for this source is A4-sized (210mm x 297mm).

***A5:** The paper for this source is A5-sized (148 x 210mm).

***B5:** The paper for this source is B5-sized (182 x 257mm).

***NONE:** No paper source number two is specified.

ENVELOPE
Specifies, for host print transform enabled printers, the type of envelopes used.

***SAME:** The value does not change.

***MFRTYPMDL:** The system uses the suggested setting for this printer.

***MONARCH:** The envelopes for this source are monarch-sized (3.875 x 7.5 inches).

***NUMBER9:** The envelopes for this source are number 9-sized (3.875 x 8.875 inches).

***NUMBER10:** The envelopes for this source are number 10-sized (4.125 x 9.5 inches).

***B5:** The envelope for this source is B5-sized (176 x 250mm).

***C5:** The envelopes for this source are C5-sized (162mm x 229mm).

***DL:** The envelopes for this source are DL-sized (110mm x 220mm).

***NONE:** No envelope source is specified.

ASCII899
Specifies, for host print transform enabled printers, whether the printer has ASCII code page 899 installed.

***SAME:** The value does not change.

***NO:** The printer does not have ASCII code page 899 installed.

***YES:** The printer does have ASCII code page 899 installed.

TEXT
Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

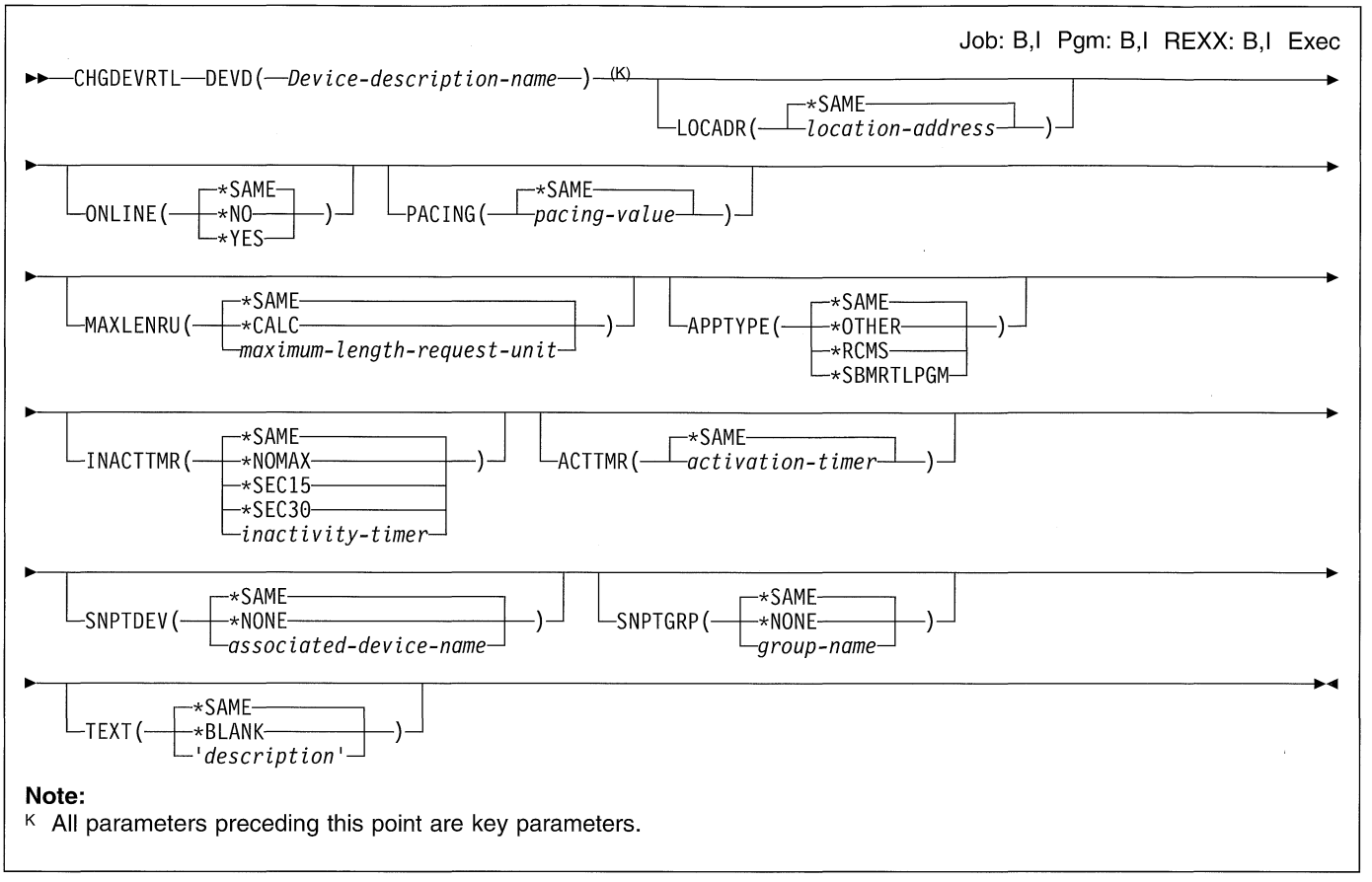
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGDEVPRT  DEVD(PRT1)  PORT(1)  SWTSET(5)
```

This command changes the device description for the printer named PRT1 so that it is now located at port 1, and its address is 5.

CHGDEVRTL (Change Device Description (Retail)) Command



Note:
K All parameters preceding this point are key parameters.

Purpose

The Change Device Description (Retail) (CHGDEVRTL) command changes the description of a retail device.

Required Parameter

DEVD
Specifies the name of the device description being changed.

Optional Parameters

LOCADR
Specifies the local location address. Specify a 2-character hexadecimal value ranging from 01 through FE.

***SAME:** The value does not change.

location-address: Specify the location address.

ONLINE
Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***NO:** This device is not automatically varied on at IPL.

***YES:** The device is automatically varied on at IPL.

PACING
Specifies a value, ranging from 0 through 7, to indicate the pacing count used in communicating with this retail device.

***SAME:** The value does not change.

pacing-value: Specify a value, ranging from 0 through 7, to indicate the pacing count used with this device.

MAXLENRU
Specifies the default maximum request unit (RU) length that can be sent or received by the local systems if the maximum size is not specified in the bind.

Note: If LOCADR is 01, then MAXLENRU must be *CALC or 256.

***SAME:** The value does not change.

***CALC:** The system determines the value to use.

maximum-length-request-unit: Specify either 247, 256, 503, 512, 1015, or 1024 bytes for the maximum length of incoming request units.

APPTYPE
Specifies the application type being used.

***SAME:** The value does not change.

***OTHER:** The retail device communicates with either the host command processor(HCP), if 01 is specified on the LOCADR address, or it communicates with an application on the controller.

***RCMS:** The retail device communicates with the remote change management server (RCMS).

***SBMRTLPGM:** The retail device is used with the Submit Retail Program (SBMRTLPGM) command to start a program on the Advanced Data Communications for Stores Start User Program (ADCS SUP) support. This value is valid only when the AS/400 Retail Point-of-Sale Communications Facility licensed program is being used on the controller.

INACTTMR

Specifies an inactivity timer (time-out) value. This parameter also specifies what happens when the time-out value is exceeded, dependent on other attributes of the device:

- For devices connected using SNA pass-through (SNPT) support, the user is informed by a message to QSYSOPR and the session is ended when the amount of time that the device is not bound to a host application exceeds the time-out value. The user must reestablish the connection and session.
- For all other attachments, valid values range from 1 through 30 minutes.

Note: This timer is not used by devices allocated to a subsystem (normal interactive use) because the subsystem always has a file open for the device. The timer is used by batch jobs that open and close files for the device.

***SAME:** The value does not change.

***NOMAX:** Maximum inactivity time is not tracked.

***SEC15:** A 15-second time-out period is used.

***SEC30:** A 30-second time-out period is used.

inactivity-timer: Specify a time-out value.

ACTTMR

Specifies, for switched lines, the amount of time (in seconds) that the SNA pass-through support waits for the device to respond to the activation request from the host AS/400 system. If the device does not respond within this time, it is considered not available.

***SAME:** The value does not change.

activation-timer: Specify a number ranging from 1 through 2550 indicating the number of seconds before the device is considered not available.

SNPTDEV

Specifies the name of the associated SNA pass-through device that is attached to a host or advanced program-to-program communications (APPC) controller.

***SAME:** The value does not change.

***NONE:** No name is specified.

associated-device-name: Specify the name of a device that is attached to a host or an APPC controller that is associated with this device.

SNPTGRP

Specifies the name configured for a group of host devices in a configuration list. This indicates that this device is associated with any one of the devices in that group which is available.

***SAME:** The value does not change.

***NONE:** No name is specified.

group-name: Specify the name configured for a group of host devices that must be associated with this device.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

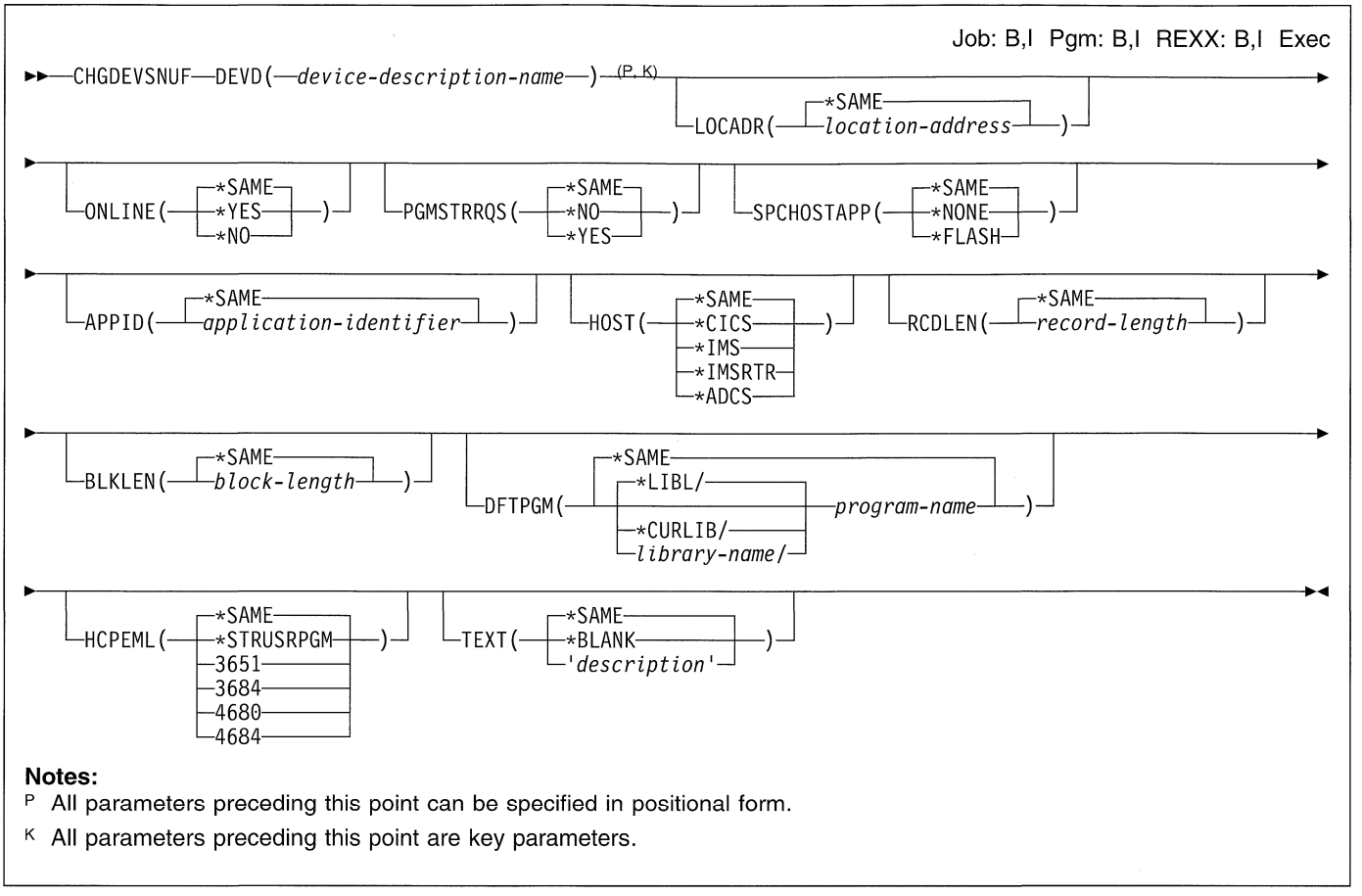
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGDEVRTL   DEVD(RTL1)  PACING(5)
```

This command changes a retail device description named RTL1, specifying 5 as the new pacing value.

CHGDEVSNUF (Change Device Description (SNUF)) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Device Description (SNUF) (CHGDEVSNUF) command changes the description of a Systems Network Architecture Upline Facility (SNUF) device. The changes are implemented the next time the device is varied on.

Required Parameter

DEVD

Specifies the name of the device description being changed.

Optional Parameters

LOCADR

Specifies the local location address. Specify a 2-character hexadecimal value ranging from 01 through FF.

***SAME:** The value does not change.

location-address: Specify two hexadecimal characters ranging from 01 through FF for the local address of the SNUF devices.

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The device is automatically varied on at IPL.

***NO:** This device is not automatically varied on at IPL.

PGMSTRRQS

Specifies whether this device is reserved for host system call by use of the Program Start Request (PGMSTRRQS) parameter.

***SAME:** The value does not change.

***NO:** This device is not reserved for call by use of the PGMSTRRQS parameter.

***YES:** This device is reserved for call by use of the PGMSTRRQS parameter.

SPCHOSTAPP

Specifies whether SNUF customizes support for special host applications outside the Customer Information Control System for Virtual Storage (CICS/VS) or Information Management System for Virtual Storage (IMS/VS) application layer.

***SAME:** The value does not change.

***NONE:** SNUF does not customize support for special host applications.

***FLASH:** SNUF customizes support for the Federal Link Access for Secondary Half-sessions (*FLASH) protocol application.

APPID
Specifies the virtual telecommunications access method (VTAM) Application Identifier sent with the sign-on message.

***SAME:** The value does not change.

application-identifier: Specify the VTAM application identifier.

HOST
Specifies the type of host system with which the device communicates.

***SAME:** The value does not change.

***CICS:** The host system type is CICS/VS.

***IMS:** The host system type is IMS/VS.

***IMSRTR:** The host system type is Information Management System Ready to Receive (IMSRTR).

***ADCS:** The host system type is Advanced Data Communication for Stores (ADCS).

RCDLEN
Specifies the maximum record length allowed when communicating with this device. The maximum value for this parameter is 32767 bytes. The value must not exceed the block length value for this device.

***SAME:** The value does not change.

record-length: Specify the maximum record length (in bytes) to use with this device file. The value must be at least the size of the largest record sent. If a record is longer than the specified record length, a run time error occurs when the record is sent or received. Valid values range from 1 through 32767 bytes for SNUF communications. For BSCEL communications, the maximum record length is 8192 bytes.

BLKLEN
Specifies the maximum block length allowed when communicating with this device. The maximum value for this parameter is 32767 bytes. This value must be greater than or equal to the record length.

***SAME:** The value does not change.

block-length: Specify the maximum block length (in bytes) of records sent. The value must be at least the size of the largest record sent. Valid values range from 1 through 32767.

DFTPGM
Specifies the qualified name of the default program called if a program start request is received and no program is specified.

***SAME:** The default program does not change.

The name of the default program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the default program.

HCPEML
Specifies the Host Command Processor (HCP) emulation to be done by the AS/400 Retail Licensed Program in support of the Advanced Data Communications for Stores (*ADCS) value on the HOST parameter. HCPEML can only be specified when *ADCS is the value for the HOST parameter.

***SAME:** The value does not change.

***STRUSRPGM:** The host is running ADCS and will use this Device Description for the Start User Program (SUP) emulated session.

3651: The host is running ADCS and will use this Device Description for a 3651 HCP emulated session.

3684: The host is running ADCS and will use this Device Description for a 3684 HCP emulated session.

4680: The host is running ADCS and will use this Device Description for a 4680 HCP emulated session.

4684: The host is running ADCS and will use this Device Description for a 4684 HCP emulated session.

TEXT
Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

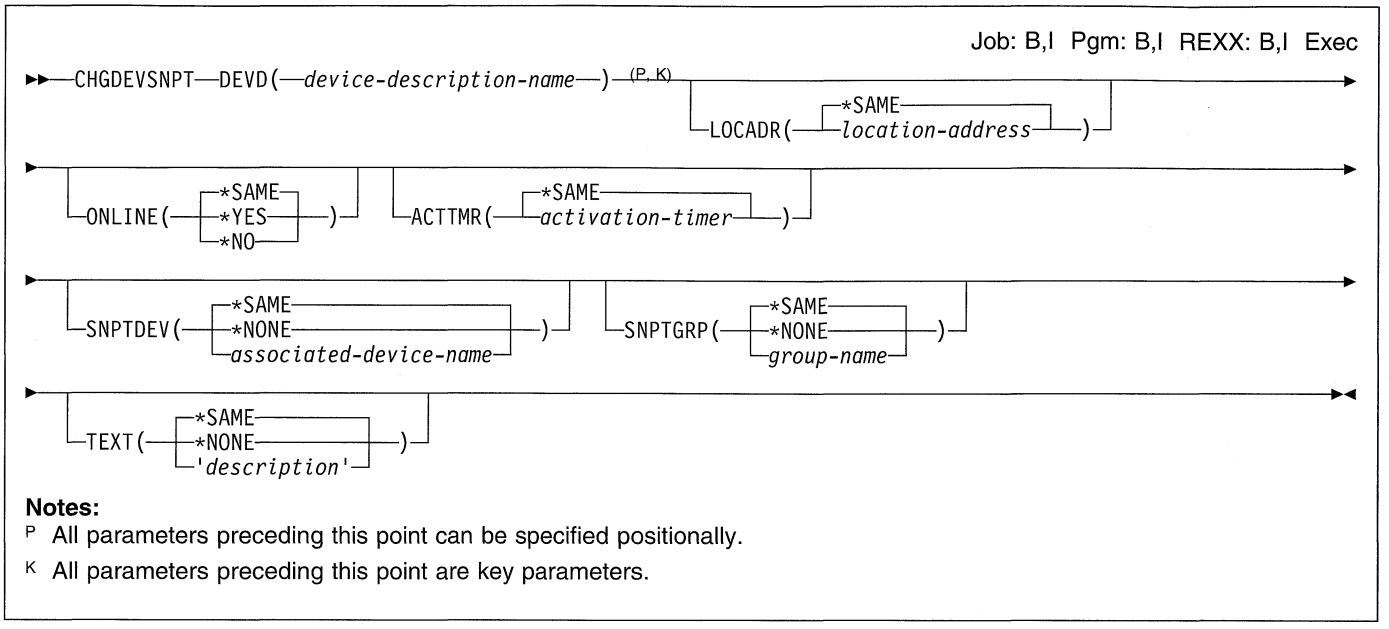
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGDEVSNUF   DEVD(SNUFDEV01)
  PGMSTRRQS(*YES)  DFTPGM(INQUIRY)
```

This command changes the device description for communications device SNUFDEV01, reserving it for host system call by the use of a program start request. The default program is changed to INQUIRY.

CHGDEVSNPT (Change Device Description (SNA Pass-Through)) Command



Notes:

P All parameters preceding this point can be specified positionally.

K All parameters preceding this point are key parameters.

Purpose

The Change Device Description (SNA Pass-Through) (CHGDEVSNPT) command changes a device description for an SNA pass-through device.

Required Parameter

DEVD

Specifies the name of the device description being changed.

Optional Parameters

LOCADR

Specifies the local location address. Specify a 2-character hexadecimal value ranging from 01 through FF.

***SAME:** The value does not change.

location-address: Specify the location address.

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The device is automatically varied on at IPL.

***NO:** This device is not automatically varied on at IPL.

ACTTMR

Specifies, for switched lines, the amount of time (in seconds) that the SNA pass-through support waits for the device to respond to the activation request from the host AS/400 system. If the device does not respond within this time, it is considered not available.

***SAME:** The value does not change.

activation-timer: Specify a number ranging from 1 through 2550 indicating the number of seconds before the device is considered not available.

SNPTDEV

Specifies the name of the associated SNA pass-through device that is attached to a host or advanced program-to-program communications (APPC) controller.

***SAME:** The value does not change.

***NONE:** No name is specified.

associated-device-name: Specify the name of a device that is attached to a host or an APPC controller that is associated with this device.

SNPTGRP

Specifies the name configured for a group of host devices in a configuration list. This indicates that this device is associated with any one of the devices in that group which is available.

***SAME:** The value does not change.

***NONE:** No name is specified.

group-name: Specify the name configured for a group of host devices that must be associated with this device.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***NONE:** No text is specified.

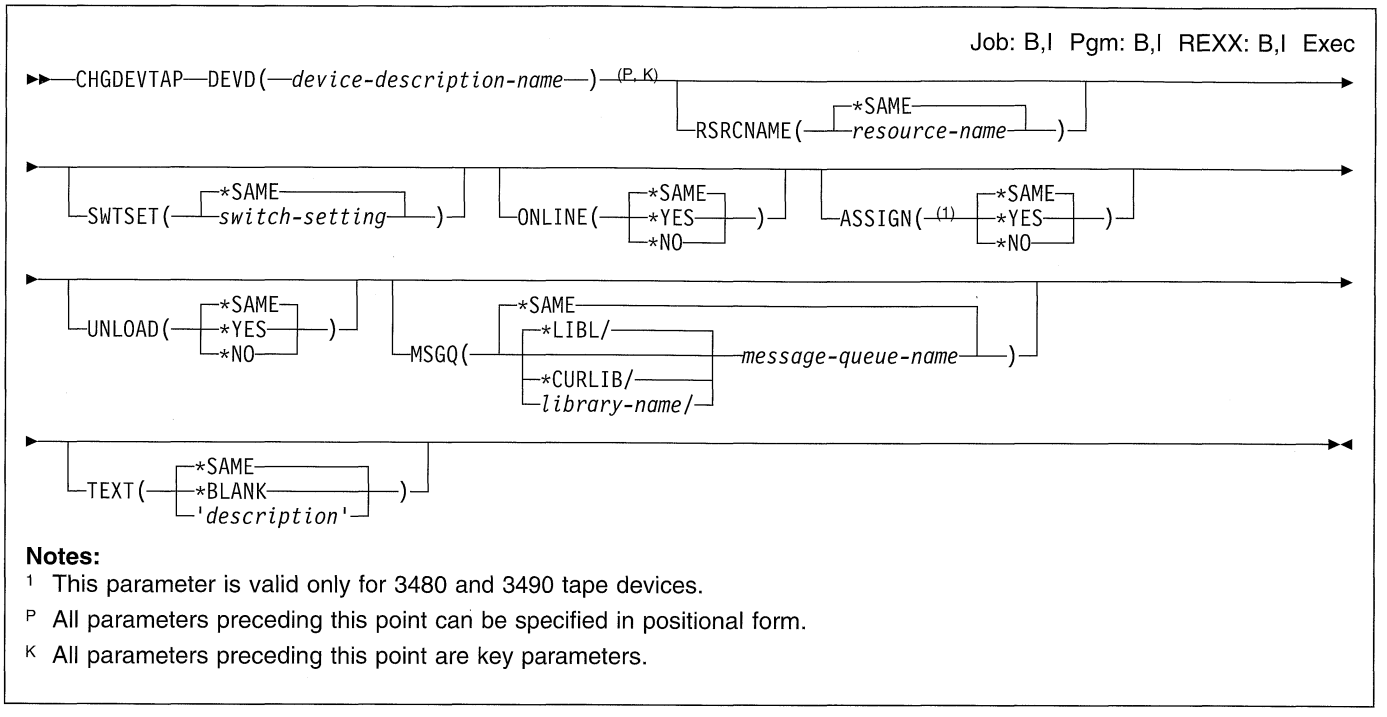
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGDEVSNPT  DEVD(SNPTDEV1)  LOCADR(05)
  SNPTDEV(DOWNDEV1)
```

This command changes an SNA pass-through device description named SNPTDEV1. The location address of the device is X'05'. The SNA pass-through device name associated with this device is DOWNDEV1.

CHGDEVTAP (Change Device Description (Tape)) Command



Notes:

1 This parameter is valid only for 3480 and 3490 tape devices.

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Device Description (Tape) (CHGDEVTAP) command changes the device description for a tape device.

Required Parameter

DEVD

Specifies the name of the device description being changed.

Optional Parameters

RSRCNAME

Specifies the resource name that describes the automatic call unit port.

Note: Use the Work with Hardware Resources (WRKHDWRSC) command with *STG specified for the TYPE parameter to determine the resource name. This parameter is not allowed for 3422, 3430, 3480, and 3490 tape devices.

***SAME:** The value does not change.

resource-name: Specify the name that identifies the physical tape device hardware on the system.

SWTSET

Specifies the switch setting or device address for this device. This parameter is only allowed for types 3422, 3430, 3480, and 3490.

***SAME:** The value does not change.

switch-setting: Specify a value ranging from 0 through F for a 3422, 3480, or 3490 tape device; specify a value ranging from 0 through 3 for a 3430 tape device.

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The device is automatically varied on at IPL.

***NO:** This device is not automatically varied on at IPL.

ASSIGN

Specifies whether the tape drive is assigned to the system when it is varied on.

***SAME:** The value does not change.

***YES:** The tape drive is assigned when the device is varied on.

***NO:** The tape drive is not assigned when the device is varied on.

UNLOAD

Specifies whether the tape drive is unloaded when the device is varied off.

***SAME:** The value does not change.

***YES:** The tape drive is unloaded when the device is varied off.

***NO:** The tape drive is not unloaded when the device is varied off. The tape is rewound, but not past the beginning-of-tape marker.

MSGQ

Specifies the qualified name of the message queue to which messages are sent.

***SAME:** The name of the message queue does not change.

The name of the message queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

message-queue-name: Specify the name of the message queue to which operational messages are sent.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

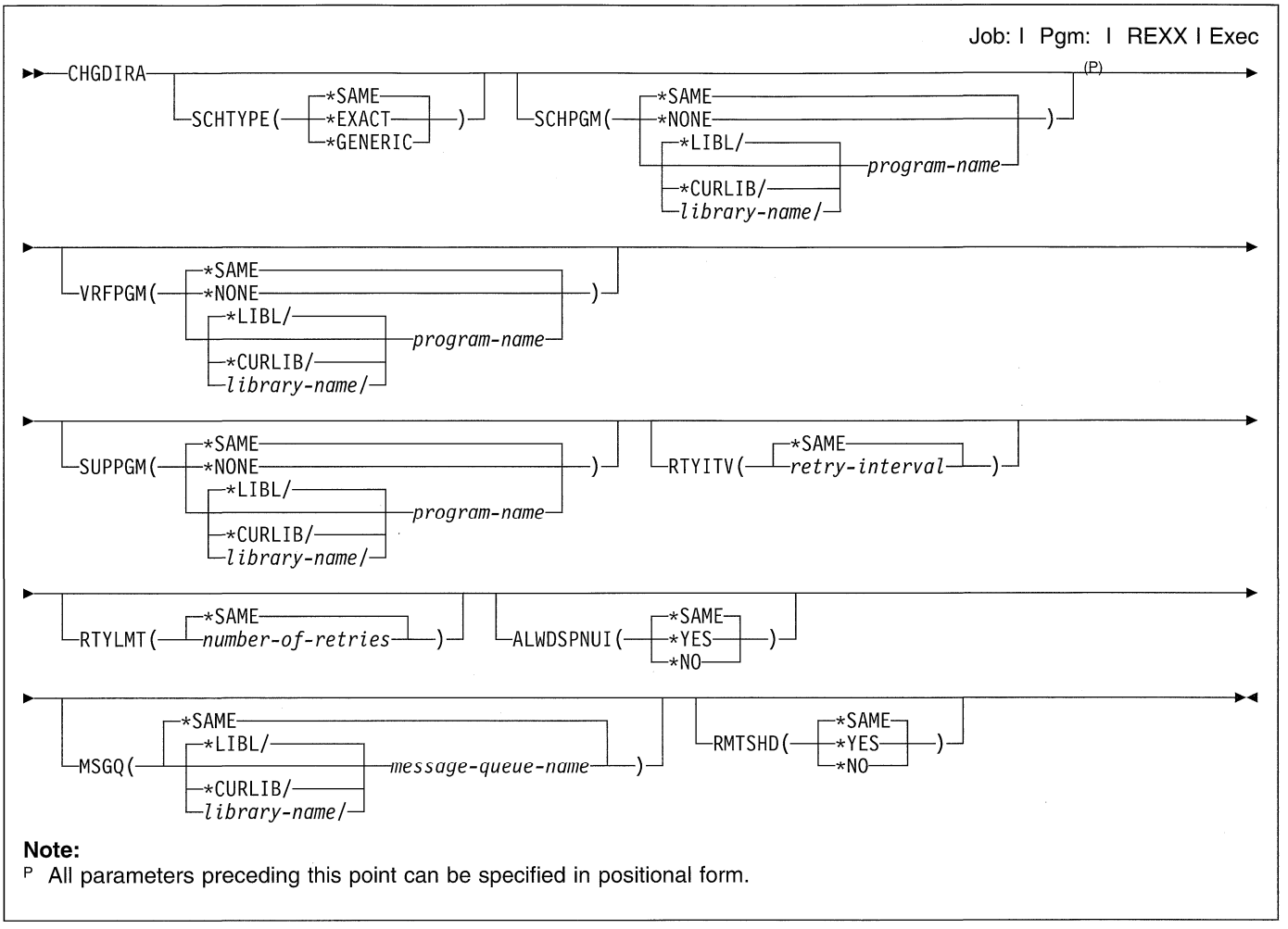
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGDEVTAP   DEVD(TAP01)  ONLINE(*YES)
```

This command changes the device description of a tape device named TAP01 so that at an IPL the device is automatically varied on.

CHGDIRA (Change Directory Attributes) Command



Note:

P All parameters preceding this point can be specified in positional form.

Purpose

The Change Directory Attributes (CHGDIRA) command allows the system administrator to set attributes used when working interactively with the directory and the directory shadow systems.

An override program is provided that fills in the values of these directory attributes.

Restrictions:

1. You must have security administrator (*SECADM) or all object (*ALLOBJ) special authority to use this command.
2. You must have all object (*ALLOBJ) special authority to change the search (SCHPGM), the verification (VRFPGM), or the supplier (SUPPGM) user exit program.

Optional Parameters

SCHTYPE

Specifies the type of search to be applied to the Search System Directory display. The search attribute specified on this parameter applies for the system.

***SAME:** The value does not change.

***EXACT:** The system searches for the exact text string specified on the Search System Directory display. This value includes the ability to specify an asterisk (*) as part of the string to find generic values.

***GENERIC:** The system searches for the text string specified on the Search System Directory display, but makes the end of the string an automatic generic search. An asterisk (*) does not need to be specified at the end of a string to find generic values.

SCHPGM

Specifies the user exit program that performs a customized search from the Search System Directory display. More information about the user exit program is in the *System Programmer's Interface Reference*.

If a user exit program is specified, it must exist.

***SAME:** The value does not change.

***NONE:** No search user exit program is specified.

The name of the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the user exit program that performs the user search.

VRFPGM

Specifies the user exit program that verifies a change, add, or delete operation for directory entries, departments, and locations that are local or shadowed. This program is called from both a local data entry and from directory shadowing. More information about the user exit program is in the *System Programmer's Interface Reference*.

Changes are always verified by the system. If a user exit program is not supplied, no additional verification checking is required by the system. When a user exit program is supplied, the user exit program is called and then system validation is performed.

If a user exit program is specified, it must exist.

***SAME:** The value does not change.

***NONE:** No authority user exit program is specified.

The name of the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the user exit program that verifies the modification.

SUPPGM

Specifies the user exit program that decides whether a change, add, or delete operation for directory entries, departments, and locations is to be shadowed to a collector system. This program is called from directory shadowing. More information about the user exit program is in the *System Programmer's Interface Reference*.

If a user exit program is not supplied, all changes are sent to the collector system. When a user exit program is supplied, the user exit program is called and then directory shadowing is performed.

If a user exit program is specified, it must exist.

***SAME:** The value does not change.

***NONE:** No authority user exit program is specified.

The name of the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the user exit program that decides which records to supply during directory shadowing.

RTYITV

Specifies the number of minutes to wait after an unsuccessful shadow before attempting to shadow again.

***SAME:** The value does not change.

retry-interval: Specify the interval (in minutes) to wait before attempting to shadow the directory data again. Valid values range from 1 through 999.

RTYLMT

Specifies the number of times to retry a directory shadow before the operation fails.

***SAME:** The value does not change.

number-of-retries: Specify the number of retries to perform before ending the directory shadow attempt. Valid values range from 0 through 9.

ALWDSPNUI

Specifies whether to allow all network user IDs to be displayed or printed by all users. The network user IDs are always displayed or printed for system administrators or for users who display or print their own directory entries.

***SAME:** The value does not change.

***YES:** All network user IDs are displayed to all users.

***NO:** Network user IDs are not displayed to all users.

MSGQ

Specifies the qualified name of the message queue to which messages are sent.

***SAME:** The value does not change.

The name of the message queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

message-queue-name: Specify the name of the message queue to which messages are sent.

RMTSHD

Specifies whether to supply additions, changes, or deletions of locally-defined remote directory entries during directory shadowing. Locally-defined remote directory entries are added locally, but have a system name that is different from the local system name. Changes are always supplied for local directory entries and for shadowed entries.

***SAME:** The value does not change.

***YES:** Additions, changes, and deletions to all directory entries are supplied to collecting systems during directory shadowing.

***NO:** Additions, changes, or deletions of locally-defined remote directory entries are not supplied during directory shadowing. Updates to local directory entries or shadowed entries are supplied to collecting systems during directory shadowing.

Examples

Example 1: Changing the Search Type to Generic

```
CHGDIRA   SCHTYPE(*GENERIC)
```

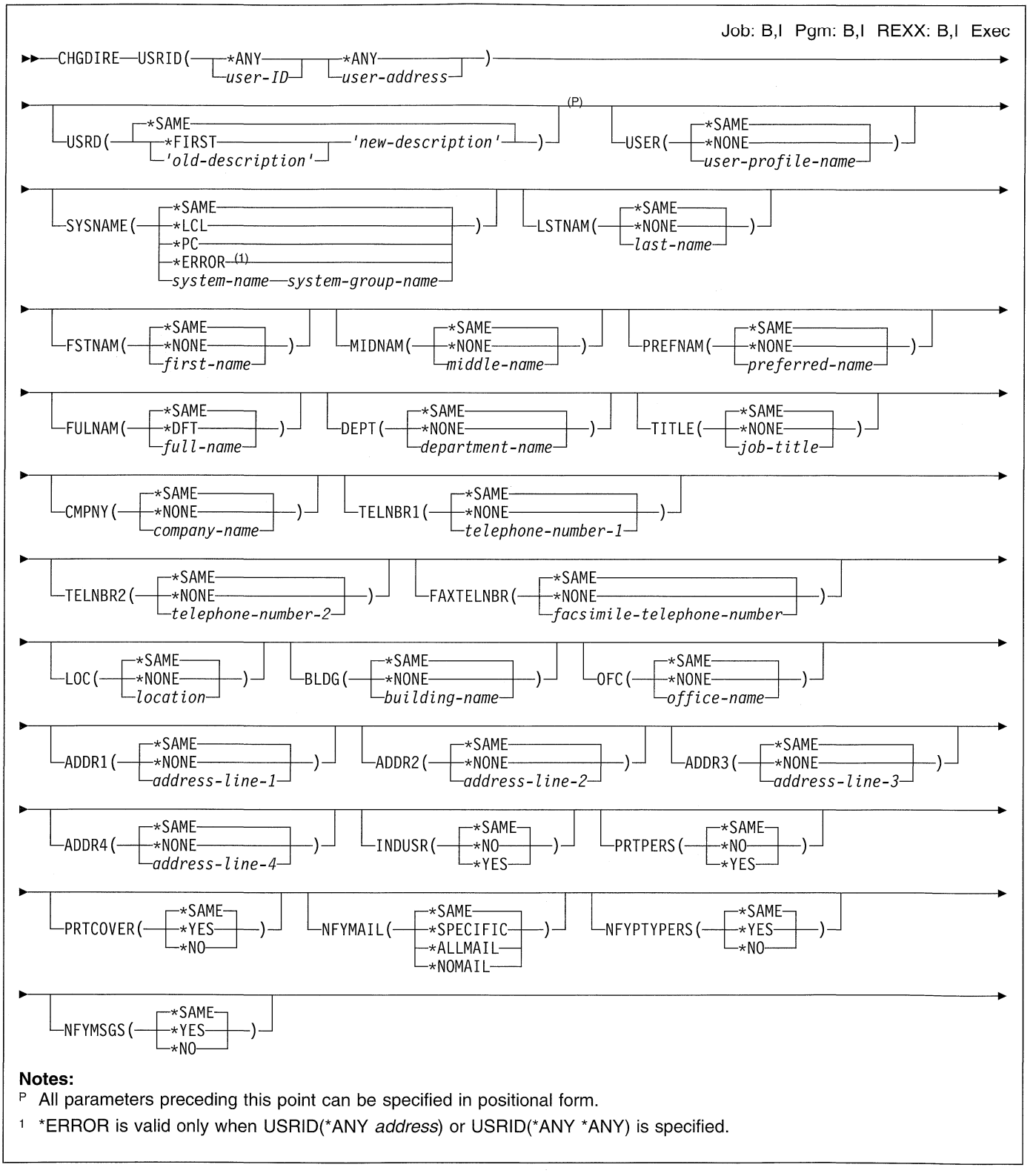
This command allows searches on the Search System Directory display to find all matches that begin with the specified text string. For example, a search for Smith may result in Smith, Smithsonian and Smithton.

Example 2: Changing the Shadowing Retry Attributes

```
CHGDIRA   RTYITV(10)  RTYLMT(3)
```

This command changes the attributes that control the available options when shadowing fails. The interval between failures is 10 minutes with a maximum of three retries for this example.

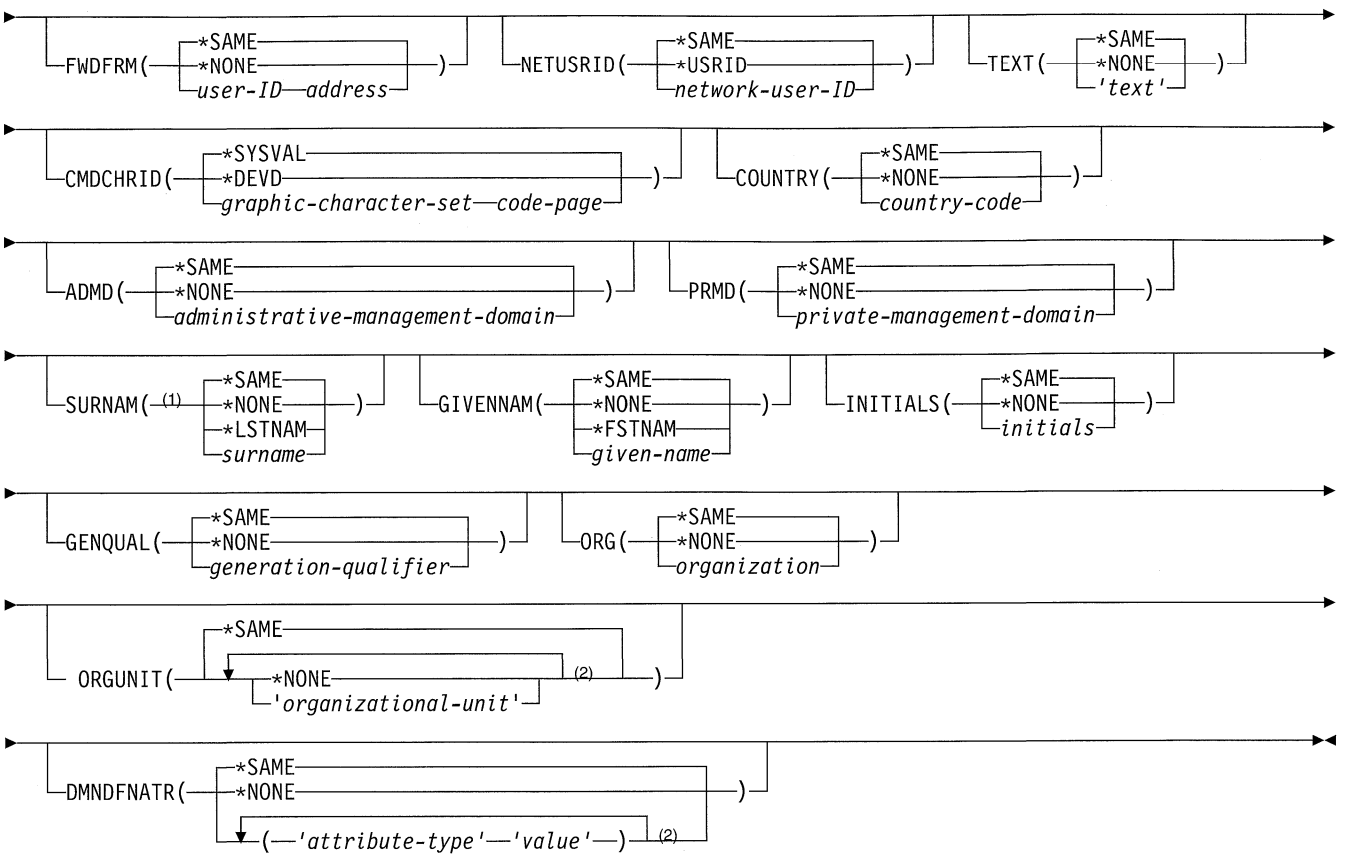
CHGDIRE (Change Directory Entry) Command



Notes:

P All parameters preceding this point can be specified in positional form.

1 *ERROR is valid only when USRID(*ANY *address*) or USRID(*ANY *ANY) is specified.



Notes:
1 SURNAM is required when GIVENAME, INITIALS, or GENQUAL is specified.
2 A maximum of 4 repetitions

Purpose

The Change Directory Entry (CHGDIRE) command changes the data for a specific entry in the system distribution directory.

This command provides support for the X.400 product (X.400). The X.400 is a limited program offering (LPO) developed for the AS/400 system as an Open System Interconnect (OSI) application. It implements the X.400 series of recommendations developed by the International Telegraph and Telephone Consultative Committee (CCITT) to allow the interchange of messages (mail) between users on systems from different vendors (for example, IBM and DEC).

The CHGDIRE command does not provide interactive display support. This is provided by the Work with Directory (WRKDIR) command.

Notes:

1. To prevent the system from changing lowercase characters to uppercase characters, enclose the values in apostrophes. This does not apply to user ID/address, system name/group, department, or X.400 originator/recipient (O/R) name.
2. Only the user ID/address, system name/group, department, and X.400 O/R name are translated from the graphic character identifier (GCID) specified by the CMDCHRID parameter. All other parameters are stored exactly as they are entered and the GCID is stored with them. The default GCID value is taken from the QCHRID system value. The user can override the defaults by specifying a character set and code page or specify *DEVD to display the device description.
3. Double-byte character set (DBCS) characters can be entered for the following system directory entry parameters:

   USRD
   LSTNAM
   FSTNAM
   MIDNAM
   PREFNAM

FULNAM
DEPT
TITLE
CMPNY
LOCATION
BLDG
OFC
ADDR1
ADDR2
ADDR3
ADDR4
TEXT

4. Administrators have authority to update any directory entry. Users who are not administrators are restricted to changing specific fields on their own directory entry. If users who are not administrators run this command and specify a USRID other than their own, an error message is returned. If a non-administrator requests a change to any of the following fields, an error message is returned, indicating the person running the command is not authorized to update these fields.

   Description
   User profile
   System name
   Indirect user
   Receiving personal mail
   Last name
   First name
   Middle name
   Preferred name
   Full name
   Department
   Forward from
   Network user ID
   X.400 O/R name fields

An X.400 O/R name in the directory can be changed with this command. X.400 is an international standard for communications and the O/R name is the addressing information used in X.400 communications. The X.400 O/R name must be in character set 1169 and code page 500. This set includes A through Z, 0 through 9, and some special characters. Additional information on characters allowed is in the *National Language Support Planning Guide*.

Restriction: The user must have administrator authority to update the directory. Users who are not administrators are restricted to updating a subset of the fields only within their own directory entry.

Required Parameter

USRID

Specifies the user ID and address of the user for whom the directory entry is changed. Both elements must be specified. If lowercase characters are specified, the system translates and stores them as uppercase characters. More information about specifying the user ID and address is in the *Distribution Services Network Guide*.

Element 1: User ID

***ANY:** Any user ID at the address specified on Element 2 of this parameter is used. Only one *ANY is allowed for each address. This value is used to resolve a distribution that does not match a specific user ID but matches an address.

user-ID: Specify the user ID for this directory entry. Up to 8 characters can be specified. If this value is specified, an address must be specified for Element 2.

Element 2: Address

***ANY:** Any address for the user ID specified on Element 1 is used. One **USRID(*ANY *ANY)** entry is allowed in the directory. This value is used to resolve distributions that do not match any other directory entries.

address: Specify the address of this directory entry. Up to 8 characters can be specified.

Optional Parameters

USRD

Specifies the description to be changed for this user and how it is changed. The description must be unique for a user ID and address; it does not need to be unique in the directory.

***SAME:** The value does not change.

Element 1: Old Description

***FIRST:** The first description for the user is changed.

'old-description': Specify the specific description to be changed.

Element 2: New Description

'new-description': Specify the new description that replaces the old description.

USER

Specifies the user profile of the user.

***SAME:** The value does not change.

***NONE:** No user profile is specified. This is valid only for remote users. If *NONE is specified for a local user, an error message is returned.

user-profile-name: Specify a valid system user profile name, which can be up to 10 characters in length. The profile name is required for all local users. If a profile name is specified for a user whose mail is sent to a remote system, the profile name must be valid.

SYSNAME

Specifies the system name and group name for the system on which the user works.

***SAME:** The value does not change.

***LCL:** The system name and group name identify the local system.

***PC:** This system name is for distributed systems node executive (DSNX) users with a personal computer (PC) attached to the system.

***ERROR:** The user's network contains a central system that receives all unresolved distributions. In this type of network, distribution looping may be encountered when a distribution cannot find a specific user ID on the intended system and the intended system has an *ANY *ANY entry directing distributions to the central system. The central system also has a default *ANY *address* entry directing unresolved distributions to the intended system. To prevent distribution looping, specify *ERROR as the system name for the entry being changed. When a distribution cannot find a specific user ID, but matches this entry, the distribution is handled as a not-valid user just as if no directory match were found.

Element 1: System Name

system-name: Specify the name of the system on which the user works.

Element 2: System Group Name

system-group-name: Specify the group name of the system on which the user works.

Note: Only the system name is required. If both the system name and group are specified, the elements must be separated by at least one space. Use up to 8 characters total for both the system name and group name. Additional information on specifying the system name is in the *Distribution Services Network Guide*.

A remote system name and group name can be assigned to a user before it is defined to the system network tables, but distributions cannot be sent to that remote user until the system name and group name are defined on the remote system. The remote system name and group are defined using the Configure Distribution Services (CFGDSTSRV) command.

LSTNAM

Specifies the user's last name.

If no names are provided (last, first, middle, preferred, or full), but a value is specified on the DEPT parameter, then the last name defaults to an asterisk (*). This is because the directory department function requires a non-blank full name when a department value is specified.

***SAME:** The value does not change.

***NONE:** The last name is changed to blanks.

last-name: Specify up to 40 characters for the user's last name.

FSTNAM

Specifies the user's first name.

***SAME:** The value does not change.

***NONE:** The first name is changed to blanks.

first-name: Specify up to 20 characters for the user's first name.

MIDNAM

Specifies the user's middle name.

***SAME:** The value does not change.

***NONE:** The middle name is changed to blanks.

middle-name: Specify up to 20 characters for the user's middle name.

PREFNAM

Specifies the user's preferred name. For example, "Jonathan" likes to be called "Jon."

***SAME:** The value does not change.

***NONE:** The preferred name is changed to blanks.

preferred-name: Specify up to 20 characters for the user's preferred name.

FULNAM

Specifies the user's full name. Directory entries are shown in the full name format when using the search and department functions. It is recommended that the user institute a consistent naming convention for the full name. Note that uppercase and lowercase alphabetic characters have different sorting sequences. Making the first character of each name uppercase and the rest that follow lowercase is the preferred format.

If FULNAM(*DFT) is specified on this field, the following format is created:

```
LAST NAME, FIRST NAME  MIDDLE NAME (PREFERRED NAME)
```

The preferred name is always enclosed in parentheses. If no user defined values are specified, but the DEPT parameter contains a value, then the last name defaults to an asterisk (*). If the user specifies FULNAM(*DFT) for the full name, it too defaults to an asterisk (*) because it is built from the last name.

***SAME:** The value does not change.

***DFT:** The full name is created from the user-defined values specified on the LSTNAM, FSTNAM, MIDNAM, and PREFNAM parameters.

full-name: Specify up to 50 characters for the user's full name.

DEPT

Specifies the name or number of the department of which the user is a member.

***SAME:** The value does not change.

***NONE:** The department name is changed to blanks. The user is no longer a member of any department.

department-name: Specify a maximum of 10 characters for the name of the user's department.

TITLE

Specifies the user's job title.

***SAME:** The value does not change.

***NONE:** The job title is changed to blanks.

job-title: Specify up to 40 characters for the user's job title.

CMPNY
Specifies the name of the company for which the user works.

***SAME:** The value does not change.

***NONE:** The company name is changed to blanks.

company-name: Specify up to 50 characters for the name of the user's company.

TELNBR1
Specifies the primary telephone number of the user. The telephone number can be specified in any arrangement appropriate for the user, including an international telephone number.

***SAME:** The value does not change.

***NONE:** The primary telephone number is changed to blanks.

telephone-number-1: Specify up to 26 characters for the primary telephone number of the user.

TELNBR2
Specifies a second telephone number for the user. The telephone number can be specified in any arrangement appropriate for the user, including an international telephone number.

***SAME:** The value does not change.

***NONE:** The second telephone number is changed to blanks.

telephone-number-2: Specify up to 26 characters for the second telephone number of the user.

FAXTELNBR
Specifies a facsimile telephone number for the user. The facsimile telephone number can be specified in any arrangement appropriate for the user, including an international telephone number.

***SAME:** The value does not change.

***NONE:** The facsimile telephone number is changed to blanks.

facsimile-telephone-number: Specify up to 32 characters for the user facsimile telephone number.

LOC
Specifies the location of the user. For example, the location can specify a building and floor, a department, or a remote site.

***SAME:** The value does not change.

***NONE:** The user location is changed to blanks.

location: Specify up to 40 characters for the location of the user.

BLDG
Specifies the name of the building in which the user works.

***SAME:** The value does not change.

***NONE:** The building name is changed to blanks.

building-name: Specify up to 20 characters for the name of the building in which the user works.

OFC
Specifies the name of the office in which the user works.

***SAME:** The value does not change.

***NONE:** The office name is changed to blanks.

office-name: Specify up to 16 characters for the name of the office in which the user works.

ADDR1-ADDR4
Specifies the mailing address of the user. Up to 40 characters of data can be entered into each of these address line fields.

***SAME:** The address line does not change.

***NONE:** The address line is changed to blanks.

address-line: Specify the user's mailing address in any arrangement up to 40 characters per line.

INDUSR
Specifies whether the user is an indirect user. An indirect user is a local user who does not sign on the system to receive electronic mail, but receives printed mail. An indirect user is a local user and must have a profile on the local system.

***SAME:** The value does not change.

***NO:** The user is not an indirect user.

***YES:** The user is an indirect user.

PRTPERS
Specifies whether an indirect user has personal mail printed. Consideration should be given to restricting public access to the printer when personal mail is printed.

When personal mail is sent to an indirect user who has specified that personal mail is not printed, the distribution is canceled, and the sender is given the reason for the cancellation. The indirect user receives a portion of the cover page indicating that personal mail was sent, but no personal mail is printed.

***SAME:** The value does not change.

***NO:** No personal mail is printed for the indirect user.

***YES:** Personal mail is printed for the indirect user.

PRTCOVER
Specifies whether a cover page is printed when the user's mail is printed.

***SAME:** The value does not change.

***YES:** The cover page is printed when a mail item is printed.

***NO:** The cover page is not printed when a mail item is printed.

NFYMAIL

Specifies whether the user is notified of the arrival of mail. The notification is a message on the user's message queue.

***SAME:** The value does not change.

***SPECIFIC:** The user is notified of the arrival of specific types of mail. The types of mail are specified on the NFYPTYPERS parameter and the NFYMSGS parameter.

***ALLMAIL:** The user is notified of the arrival of all types of mail.

***NOMAIL:** The user is not notified of the arrival of mail.

NFYPTYPERS

Specifies whether the user is notified of the arrival of priority and personal mail. This parameter is valid only if NFYMAIL(*SPECIFIC) is specified.

***SAME:** The value does not change.

***YES:** The user is notified of the arrival of priority and personal mail.

***NO:** The user is not notified of the arrival of priority and personal mail.

NFYMSGS

Specifies whether the user is notified of the arrival of messages. This parameter is valid only if NFYMAIL(*SPECIFIC) is specified.

***SAME:** The value does not change.

***YES:** The user is notified of the arrival of messages.

***NO:** The user is not notified of the arrival of messages.

FWDFRM

Specifies whether distributions are automatically forwarded from a specified user ID and address. This value is valid only for local users. This value cannot be an existing user ID, address, or forward-from value in the directory. It is used with the Rename Directory Entry (RNMDIRE) command to allow distributions to be sent to the old user until all users can be renamed or changed.

***SAME:** The value does not change.

***NONE:** Distributions are not forwarded.

Element 1: User ID

user-ID: Specify the user ID from which distributions are to be forwarded. A maximum of 8 characters can be specified. If this value is specified, an address must be specified on Element 2.

Element 2: Address

address: Specify the address from which distributions are to be forwarded. A maximum of 8 characters can be specified.

NETUSRID

Specifies the network user ID for the directory entry. The network user ID is used to identify a user in a network.

***SAME:** The value does not change.

***USRID:** Change the network user ID to the user ID and address associated with this entry. The format of the network user ID is the 8 character user ID, 1 blank character, and the 8-character address.

network-user-ID: Specify the network user ID for this user. A maximum of 47 characters can be specified.

TEXT

Specifies text that briefly describes the directory entry. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***NONE:** The text is changed to blanks.

'text': Specify up to 50 characters of text to describe additional information about the user.

CMDCHRID

Specifies the character identifier (graphic character set and code page) for data being specified as parameter values on this command. This character identifier (CHRID) is related to the display device used to specify the command. More information about CHRID processing is in the *Guide to Programming Displays*.

***SYSVAL:** The system determines the graphic character set and code page values for the command parameters from the QCHRID system values.

***DEVD:** The system determines the graphic character set and code page values for the command parameters from the display device description where this command is entered. This option is supported only when the command is entered from an interactive job. If this option is specified in a batch job, an error message is returned.

Element 1: Character Set

graphic-character-set: Specify the character set used to create the command parameters. Valid values range from 1 through 9999 characters.

Element 2: Code Page

code-page: Specify the code page. Valid values range from 1 through 9999.

COUNTRY

Specifies the country name part of the X.400 O/R name.

***SAME:** The value does not change.

***NONE:** The country is changed to blanks.

country-code: Specify an ISO 3166 Alpha-2 code or a CCITT country code. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

ADMD
Specifies the administrative management domain part of the X.400 O/R name.

***SAME:** The value does not change.

***NONE:** The administrative management domain is changed to blanks.

administrative-management-domain: Specify a 1- to 16-character administrative management domain name. An administrative management domain is a public organization that handles a management domain. A management domain is a set of message transfer agents and user agents that comprise a system capable of handling messages.

PRMD
Specifies the private management domain part of the X.400 O/R name.

***SAME:** The value does not change.

***NONE:** The private management domain is changed to blanks.

private-management-domain: Specify a 1- to 16-character private management domain name. A private management domain is a private company or non-commercial organization that handles a management domain. A management domain is a set of message transfer agents and user agents that comprise a system capable of handling messages.

SURNAM
Specifies the X.400 user last name part of the personal name within the X.400 O/R name.

***SAME:** The value does not change.

***NONE:** The surname is changed to blanks.

***LSTNAM:** The user last name specified in the directory entry is used as the surname.

surname: Specify a 1- to 40-character surname.

GIVENNAM
Specifies the X.400 user given (first) name part of the personal name within the X.400 O/R name.

***SAME:** The value does not change.

***NONE:** The given name is changed to blanks.

***FSTNAM:** The user first name specified in the directory entry is used as the given name. It is truncated to 16 characters.

given-name: Specify 1 to 16 characters for the given name.

INITIALS
Specifies the initials part of the personal name within the X.400 O/R name.

***SAME:** The value does not change.

***NONE:** The initials are changed to blanks.

initials: Specify 1 to 5 characters for the initials.

GENQUAL
Specifies the generation qualifier part of the personal name within the X.400 O/R name.

***SAME:** The value does not change.

***NONE:** The generation qualifier is changed to blanks.

generation-qualifier: Specify 1 to 3 characters for the generation qualifier.

ORG
Specifies the organization name part of the X.400 O/R name.

***SAME:** The value does not change.

***NONE:** The organization name is changed to blanks.

organization: Specify a 1- to 64-character organization name.

ORGUNIT
Specifies the organization-defined unit part of the X.400 O/R name.

***SAME:** The value does not change.

***NONE:** The organizational unit is changed to blanks.

'organizational-unit': Specify the 1- to 32-character name of an organizational unit, enclosed in apostrophes (for example, 'division' or 'department'). Up to 4 organizational units may be listed in order of descending significance.

DMNDFNATR
Specifies the type and value of a domain-defined attribute not specified by X.400 standards but allowed in the X.400 O/R name to accommodate existing systems of sending messages. Up to 4 attributes can be specified.

***SAME:** The value does not change.

***NONE:** The domain-defined attribute is changed to blanks.

Element 1: Type

'attribute-type': Specify a 1- to 8-character type description for the domain-defined attribute, enclosed in apostrophes.

Element 2: Value

'value': Specify a 1- to 128-character value description for the domain-defined attribute, enclosed in apostrophes.

Examples

Example 1: Changing a User's Telephone Number

```
CHGDIRE   USRID(HURST PAYROLL)  USER(*SAME)
  TELNBR1('456-4489')
```

Assume the user who runs the command has user ID HURST PAYROLL and does not have security administrator authority. The primary telephone number for the user is changed. All other information remains the same. This command works the same if someone other than HURST PAYROLL runs the command and has security administrator authority. If the person running the command is not HURST PAYROLL and does not have security administrator authority, an error message is returned.

Example 2: Changing a User's Address, Telephone Number, and Text Information

```
CHGDIRE   USRID(BYRD NEWYORK)  USER(AJBYRD)
  SYSNAME(*LCL) ADDR1('Dept55N/025-3')
  ADDR2('IBM Rochester') ADDR3(*NONE)
  ADDR4(*NONE) TELNBR1('456-4489')
  LOC(Rochester) TEXT('User transferred from Boca')
```

Assume the person running this command has security administrator authority. The user (BYRD NEWYORK) has transferred from a remote user to a local user. The profile name must now be specified since the user is now a local user. The user's address, telephone number, and text information are updated.

Example 3: Changing a User's Full Name, Department, and Office

```
CHGDIRE   USRID(JANE CHICAGO)  LSTNAM('Smith')
  MIDNAM('Allen')  FULNAM(*DFT)  DEPT(55N)
  OFC(L305)
```

Assume the person running this command has security administrator authority. The user JANE CHICAGO has changed her name and at the same time has moved to a new office and department. Jane's full name is changed to 'Smith, Jane Allen'. If FULNAM(*SAME) is specified or is used as the default, Jane's full name remains the same, even though her first and middle names are changed.

Additional Considerations

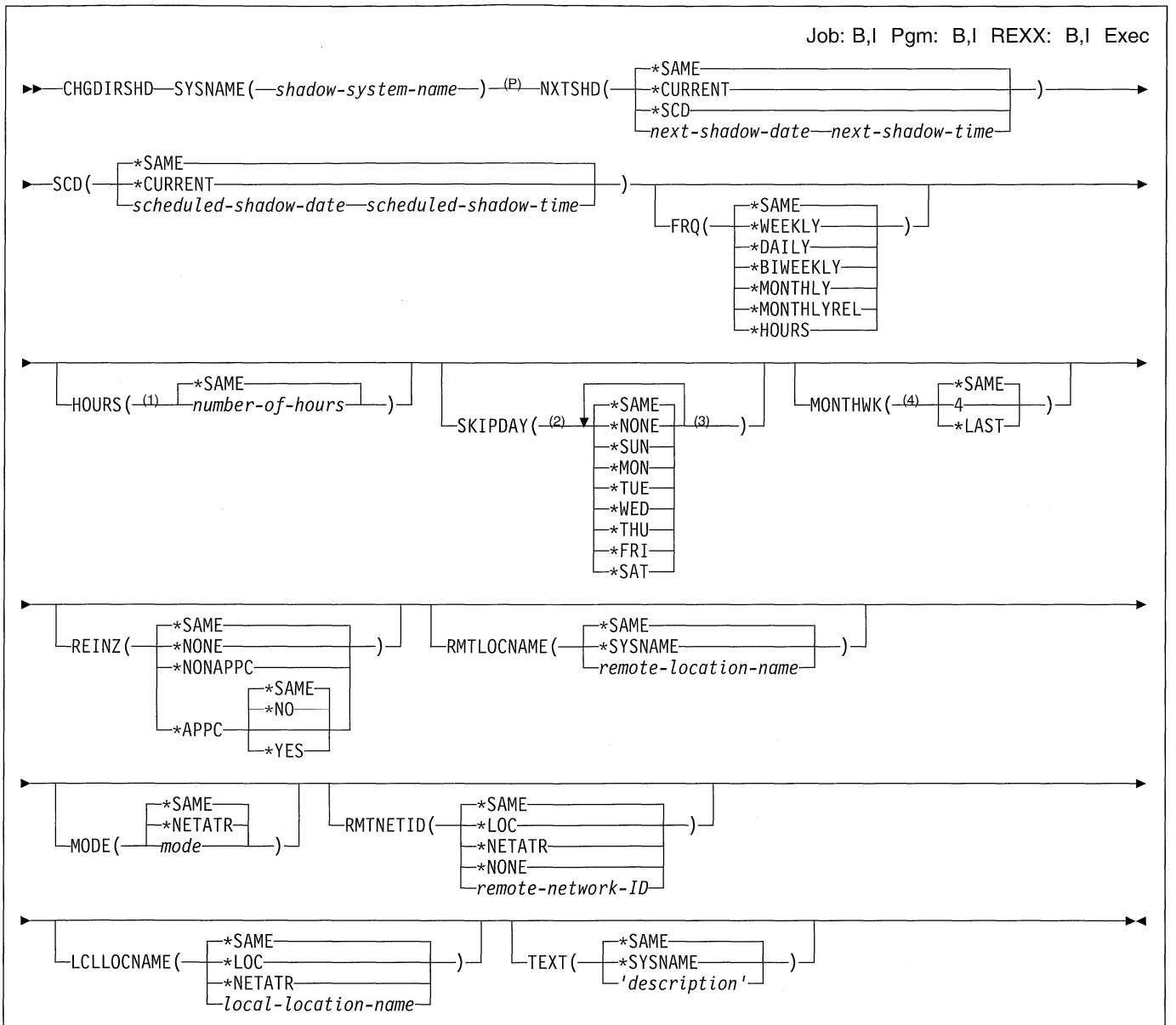
The user ID and address fields can be changed using the Rename Directory Entry (RNMDIRE) command.

The user description field can be changed using this command. A consistent use of the description field is helpful when using the *sort by description* function from the Work with Directory display.

For local users, there is a one-to-one correspondence between the user ID and address and the user profile. Only one user ID and address is associated with a user profile name, and only one user profile name is associated with a user ID and address. If a user profile name is specified on the CHGDIRE command that is already associated with an existing user ID in the directory, an error message is returned. For remote users, this should not cause a problem since the user profile name should not be specified. However, if a profile name is specified, it is verified to determine that the profile name is not already in the directory. If the profile is in the directory, an error message is returned.

Administrators have authority to update any directory entry. Users who are not administrators are restricted to updating only specific fields for their own directory entry. If users who are not administrators run this command and specify a USRID other than their own, an error message is returned. If a non-administrator requests changes to the user profile, system name, indirect user, or receiving of personal mail, an error message is returned, indicating the person running the command is not authorized to update these fields.

CHGDIRSHD (Change Directory Shadow System) Command



Notes:

1 Valid only when FRQ(*HOURS) is specified.

2 Valid only when when FRQ(*DAILY) is specified.

3 A maximum of 5 repetitions.

4 Valid only when FRQ(*MONTHLYREL) is specified and the date specified on the SCD parameter is the 22nd, 23rd or 24th.

P All parameters preceding this point can be specified in positional form.

Purpose

The Change Directory Shadow System (CHGDIRSHD) command changes a system that is defined to be shadowed.

Restriction: To use this command, you must have security administrator (*SECADM) authority.

Required Parameters

SYSNAME

Specifies the system that supplies data to the local system. A maximum of 8 characters can be specified for the name of the supplier system you are changing.

You can specify uppercase letters A through Z, numbers 0 through 9, and special characters @, #, $, and embedded blanks. Embedded blanks must be enclosed in single quotation marks ('). Leading blanks are not allowed. The @, #, and $ characters are not recommended because they are not part of an invariant character set and are not available on all keyboards.

NXTSHD

Specifies the date and time when the next shadow occurs. This is the same as the scheduled shadow date unless a retry is performed or a next shadow date is specified in addition to the scheduled shadow date.

***SAME:** The value does not change.

***CURRENT:** The current date and time is used.

***SCD:** The scheduled date and time is used.

Element 1: Next Shadow Date

next-shadow-date: Specify the date on which the system next begins shadowing data to your system. The date must be specified in the job date format.

Element 2: Next Shadow Time

next-shadow-time: Specify the time at which the system next begins shadowing data to your system.

The time is specified in 24-hour format with or without a time separator as follows:

- With a time separator, specify a string of 5 or 8 digits where the time separator separates the hours, minutes, and seconds. If this command is entered from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command fails.
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds. Valid values for **hh** range from 00 through 23. Valid values for **mm** and **ss** range from 00 through 59.

SCD

Specifies the date and time of the scheduled shadow.

***SAME:** The value does not change.

***CURRENT:** The system begins shadowing data at the current date and time.

Element 1: Shadow Date

scheduled-shadow-date: Specify the date on which the system begins shadowing data to your system. The date must be specified in the job date format.

Element 2: Shadow Time

scheduled-shadow-time: Specify the time at which the system begins shadowing data to your system.

The time is specified in 24-hour format with or without a time separator as follows:

- With a time separator, specify a string of 5 or 8 digits where the time separator separates the hours, minutes, and seconds. If this command is entered from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command fails.
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds. Valid values for **hh** range from 00 through 23. Valid values for **mm** and **ss** range from 00 through 59.

Optional Parameters

FRQ

Specifies the frequency with which the supplier system you are changing shadows data to your system, based on the value specified on the SCD parameter.

***SAME:** The value does not change.

***WEEKLY:** Shadowing occurs once a week.

***DAILY:** Shadowing occurs once a day.

***BIWEEKLY:** Shadowing occurs every other week.

***MONTHLY:** Shadowing occurs on the same date every month.

***MONTHLYREL:** Shadowing occurs on the same relative day of the same relative week of every month, such as the first Monday of the month.

***HOURS:** Shadowing occurs in the interval specified on the HOURS parameter.

HOURS

Specifies the number of hours between shadows from the supplier system you are changing.

***SAME:** The value does not change.

number-of-hours: Specify the number of hours between shadowing.

SKIPDAY

Specifies, when FRQ(*DAILY) is specified, the days of the week when shadowing does not occur. A maximum of five values, other than *NONE, can be specified.

***SAME:** The value does not change.

***NONE:** No days are skipped.

***SUN:** Sundays are skipped.

***MON:** Mondays are skipped.

***TUE:** Tuesdays are skipped.

***WED:** Wednesdays are skipped.

***THU:** Thursdays are skipped.

***FRI:** Fridays are skipped.

***SAT:** Saturdays are skipped.

MONTHWK

Specifies whether shadowing that occurs on the same relative day of the month is scheduled to occur in the fourth week or the last week of the month.

***SAME:** The value does not change.

4: Shadowing occurs on the same relative day in the fourth week of the month.

***LAST:** Shadowing occurs on the same relative day in the last week of the month, whether or not the month has four or five weeks.

REINZ

Specifies the method used if the first shadow is done again. The first shadow duplicates all of the data in the supplier system's distribution directory. Subsequent shadows include only data that has changed since the previous shadow.

***SAME:** The value does not change.

***NONE:** The shadowed directory data is not reinitialized.

***NONAPPC:** The Copy to Directory (CPYTODIR) command is used to reinitialize the directory.

Element 1: Method

***APPC:** The first shadow occurs when this command is run using advanced program-to-program (APPC) communications. If you are adding a supplier system with a large directory, you may want to specify *NONAPPC to prevent the first shadow from monopolizing your communications lines.

When *APPC is specified, Element 2 allows you to specify whether the data in the fields of a directory entry on your system is replaced by shadowed data if the same entry also exists in the supplier system's directory.

Element 2: Replace Data

***SAME:** The value does not change.

***NO:** The data in the fields of existing directory entries on your system is not replaced with data from the supplier system.

***YES:** All shadowed data is added to your system distribution directory. The data in the fields of existing directory entries on your system is replaced with shadowed data if the same entry also exists in the supplier system's directory.

RMTLOCNAME

Specifies the remote location name of the supplier system you are changing.

***SAME:** The value does not change.

***SYSNAME:** The value specified on the SYSNAME parameter is used for the remote location name.

remote-location-name: Specify the full name of a remote location.

A maximum of 8 characters can be specified. The first character must be an uppercase letter A through Z, or special character $, #, or @. The name cannot contain a blank, plus sign (+), period (.), or an underscore (_). For more information, see the *APPC Programmer's Guide*.

MODE

Specifies the name of the mode that defines the sessions on the device used when shadowing data from the supplier system.

***SAME:** The value does not change.

***NETATR:** The mode name specified in the network attributes is used.

mode-name: Specify the mode name.

A maximum of 8 characters can be specified. The first character must be an uppercase letter A through Z, or special character $, #, or @. The name cannot contain a blank, plus sign (+), period (.), or an underscore (_). For more information, see the *APPC Programmer's Guide*.

RMTNETID

Specifies the supplier system's remote network identifier (ID).

***SAME:** The value does not change.

***LOC:** The remote network identifier (ID) associated with the remote location is used. If several remote network IDs are associated with the remote location, the system determines which remote network ID is used.

***NETATR:** The RMTNETID value specified in the system network attributes is used.

***NONE:** No remote network identifier (ID) is used.

remote-network-ID: Specify the remote network ID.

A maximum of 8 characters can be specified. The first character must be an uppercase letter A through Z, or special character $, #, or @. The name cannot contain a blank, plus sign (+), period (.), or an underscore (_). For more information, see the *APPC Programmer's Guide*.

LCLLOCNAME

Specifies the local location name. The local location name is used to identify your system to the supplier system you are changing.

***SAME:** The value does not change.

***LOC:** The local location name associated with the remote location is used.

***NETATR:** The LCLLOCNAME value specified in the system network attributes is used.

local-location-name: Specify the local location name.

A maximum of 8 characters can be specified. The first character must be an uppercase letter A through Z, or special character $, #, or @. The name cannot contain a blank, plus sign (+), period (.), or an underscore (_). For more information, see the *APPC Programmer's Guide*.

TEXT

Specifies text that briefly describes the directory shadow system. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***SYSNAME:** The name specified on the SYSNAME parameter is used for the description.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Examples

Example 1: Changing the Next Shadow Date and Time

```
CHGDIRSHD   SYSNAME(NYCITY)
  NXTSHD( '93/05/01' '8:00:00' )
```

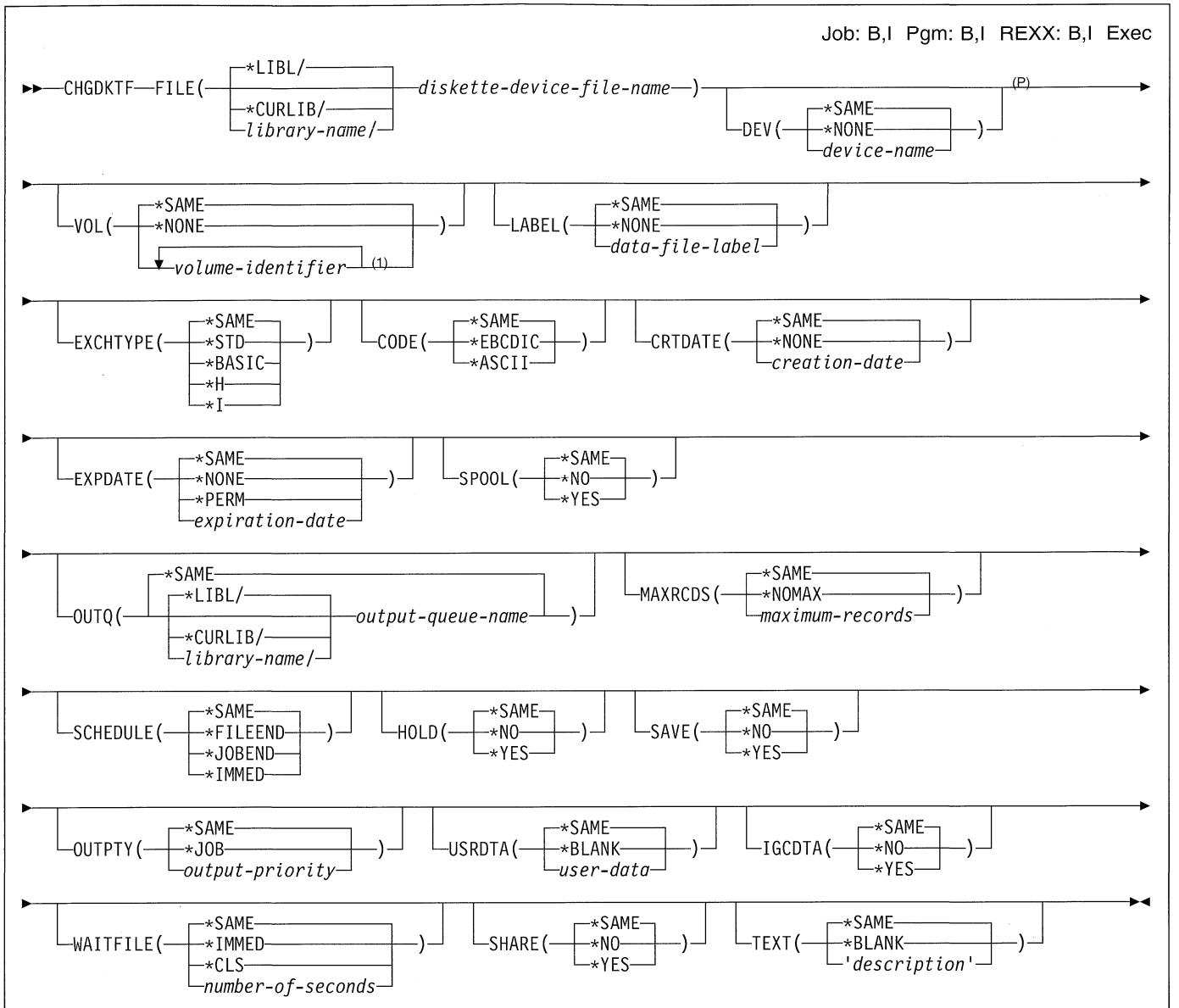
This command changes the next shadowing occurrence for the system NYCITY. This does not effect the scheduled date and time. When the shadowing is completed for this system, the next date and time will be calculated from the scheduled date and time. This command is used for temporary changes to shadowing schedules.

Example 2: Changing the Shadow Date and Time

```
CHGDIRSHD  SYSNAME(NEWYORK)
  SCD('93/06/01' '20:00:00')  FRQ(*BIWEEKLY)
```

This command changes the scheduled shadow date for system NEWYORK to June 1, 1993 at 8:00 PM. The frequency of the shadow is every 2 weeks.

CHGDKTF (Change Diskette File) Command



Notes:

P All parameters preceding this point can be specified in positional form.

1 A maximum of 50 repetitions

Purpose

The Change Diskette File (CHGDKTF) command changes, in the file description, one or more of the attributes of the specified diskette file.

Required Parameter

FILE

Specifies the qualified name of the diskette file whose description is being changed.

The name of the diskette file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

diskette-device-file-name: Specify the name of the diskette file whose description is being changed.

Optional Parameters

DEV

Specifies the name of the diskette device used with this diskette device file to perform input/output data operations. The device name of the IBM-supplied diskette device description is QDKT. This parameter is ignored if SPOOL(*YES) is specified for the file when it is opened.

***SAME:** The value does not change.

***NONE:** No device name is specified. The name of the diskette device can be specified before the device file is opened, in a Change Diskette File (CHGDKTF) command or Override with Diskette File (OVRDKTF) command, or in the high-level language program that opens the file.

device-name: Specify the name of the device that is used with this diskette device file. The device name must already exist on the system as a device description before this device file is created.

VOL

Specifies one or more volume identifiers used by the file. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***NONE:** The diskette volume identifiers are not specified for this file in this command. They can be specified later before the device file is opened, either in a Override with Diskette File (OVRDKTF) command or a Change Diskette File (CHGDKTF) command, or in the high-level language program. Otherwise, no volume identifier checking is done.

volume-identifier: Specify the identifiers of one or more volumes in the order in which they are put on the device and used. Each volume identifier contains up to 6 alphanumeric characters. A blank is used as a separator character when listing multiple identifiers.

LABEL

Specifies the data file label of the data file on diskette that is used with this diskette device file. For input files (diskette input to system), this label specifies the identifier of the file that exists on the diskette. For output files (system output to diskette), the label specifies the identifier of the file that is created on the diskette. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***NONE:** The data file label is not specified here. It can be specified before the device file is opened, in a CHGDKTF or OVRDKTF command, or in the high-level language program that opens the file.

data-file-label: Specify up to 8 characters for the identifier of the data file used with this diskette device file.

EXCHTYPE

Specifies, for diskette output files only, the exchange type used by the device file when the system is writing diskette data. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions." More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***STD:** The basic exchange format is used for a type 1 or a type 2 diskette. The H exchange type is used for a type 2D diskette.

***BASIC:** The basic exchange type is used.

***H:** The H exchange type is used.

***I:** The I exchange type is used.

CODE

Specifies the character code used. The code can be either extended binary-coded decimal interchange code (*EBCDIC) or the American National Standard Code for Information Interchange (*ASCII).

***SAME:** The value does not change.

***EBCDIC:** The extended binary-coded decimal interchange code (EBCDIC) character set code is used.

***ASCII:** The ASCII character set code is used.

CRTDATE

Specifies the date when the diskette data file was created on the diskette.

Note: The creation date parameter is valid only for diskette input data files. If the creation date written on the diskette containing the data file does not match the date specified for the device file when it is opened, an error message is sent to the user program.

***SAME:** The value does not change.

***NONE:** The creation date is not specified. It is not checked unless it is supplied before the device file is opened, either in a OVRTAPF command or CHGTAPF command, or in the high-level language program.

creation-date: Specify the creation date of the data file used by this device file. The date must be specified in the format defined by the job attributes DATFMT and, if separators are used, DATSEP. However, the specified date is put in the diskette label in the format `yymmdd`.

EXPDATE

Specifies the expiration date. The files cannot be overwritten until the expiration date. The expiration date must be later than or equal to the current date.

***SAME:** The value does not change.

***NONE:** No expiration date for the data file is specified; the file is protected for 1 day. Its protection ends the day after it is created.

***PERM:** The data file is permanently protected. An expiration date of 999999 is assigned.

expiration-date: Specify the expiration date of the data file. The date must be specified in the format defined by the job attributes DATFMT and, if separators are used, DATSEP. However, the specified date is put in the diskette label as yymmdd.

SPOOL

Specifies whether the input or output data for the diskette device file is spooled.

***SAME:** The value does not change.

***NO:** The data is not spooled. If this file is opened for input, the data is read directly from the diskette. If this is an output file, the data is written directly to the diskette as it is processed by the program.

Note: If SPOOL(*NO) is specified, the following parameters in this command are ignored: OUTQ, MAXRCDS, SCHEDULE, HOLD, SAVE, OUTPTY, and USRDTA.

***YES:** The data is spooled. If this file is opened for input, an inline data file having the specified name is processed; otherwise, the next unnamed inline spooled file is processed. More information on named and unnamed inline files is in the *Guide to Programming for Tape and Diskette*. If this is an output file, the data is spooled for processing by a diskette or print writer.

OUTQ

Specifies the qualified name of the output queue.

***SAME:** The output queue does not change.

The name of the output queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

output-queue-name: Specify the name of the output queue to which the output data is spooled. The IBM-supplied output queue that is used by the diskette file is the QDKT output queue, stored in the QGPL library.

MAXRCDS

Specifies, for spooled files only, the maximum number of records in the spooled file for spooled jobs using this diskette device file.

***SAME:** The value does not change.

***NOMAX:** The system maximum is used.

maximum-records: Specify the maximum number of diskette records that are in the spooled file. Valid values range from 1 through 500000.

SCHEDULE

Specifies, for spooled output only, when the spooled file is available to a writer.

***SAME:** The value does not change.

***FILEEND:** The spooled file is made available to the writer as soon as the file is closed in the program.

***JOBEND:** The spooled file is made available to the writer only after the entire job is completed.

***IMMED:** The spooled file is made available to the writer as soon as the file is opened in the program.

HOLD

Specifies, for spooled output only, whether the spooled file is held. The spooled file can be released by using the Release Spooled File (RLSSPLF) command.

***SAME:** The value does not change.

***NO:** The spooled printer file is not held by the output queue. The spooled output is available to a writer based on the SCHEDULE parameter value.

***YES:** The spooled file is held until released by the Release Spool File (RLSSPLF) command.

SAVE

Specifies, for spooled output only, whether the spooled file is saved (left on the output queue) after the output has been produced.

***SAME:** The value does not change.

***NO:** The spooled file data is not saved on the output queue after it has been produced.

***YES:** The spooled file data is saved on the output queue until the file is deleted.

OUTPTY

Specifies the output priority for spooled output files that are produced by this job. The highest priority is 1 and the lowest priority is 9. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***JOB:** The output priority associated with the job that created the spooled file is used.

output-priority: Specify the output priority. Valid values range from 1 (high priority) through 9 (low priority).

USRDTA

Specifies, for spooled output only, the user-specified data that identifies the file.

***SAME:** The value does not change.

***BLANK:** Ten blanks are used as the user data.

user-data: Specify up to 10 characters of text.

IGCDTA

Specifies whether the file processes double-byte character set (DBCS) data.

***SAME:** The value does not change.

***NO:** The file does not process DBCS data.

***YES:** The file processes DBCS data.

WAITFILE

Specifies the number of seconds that the program waits for the file resources and session resources to be allocated when the file is opened, or for the device or session resources to be allocated when an acquire operation is performed to the file. If those resources are not allocated within the specified wait time, an error message is sent to the program. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

Note: An immediate allocation of the device by the device resource is required when an acquire operation is performed to the file.

***SAME:** The value does not change.

***IMMED:** The program does not wait; when the file is opened, an immediate allocation of the file resources is required.

***CLS:** The job default wait time is used as the wait time for the file resources being allocated.

number-of-seconds: Specify the number of seconds that the program waits for the file resources to be allocated to the diskette file when the file is opened, or the wait time for the device allocated when an acquire operation is performed to the file. Valid values range from 1 through 32767 seconds.

SHARE

Specifies whether the open data path (ODP) for the diskette file is shared with other programs in the routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

More information on shared database files is in the *Database Guide*.

***SAME:** The value does not change.

***NO:** The ODP created by the program with this attribute is not shared with other programs in the routing step. Every time a program opens the file with this attribute, a new ODP to the file is created and activated.

***YES:** The ODP created with this attribute is shared with each program in the routing step that also specifies SHARE(*YES) when it opens the file.

Note: When SHARE(*YES) is specified and control is passed to a program, a read operation in that program retrieves the next input record. A write operation produces the next output record.

TEXT

Specifies text that briefly describes the diskette device file. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Examples

Example 1: Data is Not Spooled

```
CHGDKTF   FILE(ACCREC/PRNTRPT)  SPOOL(*NO)
```

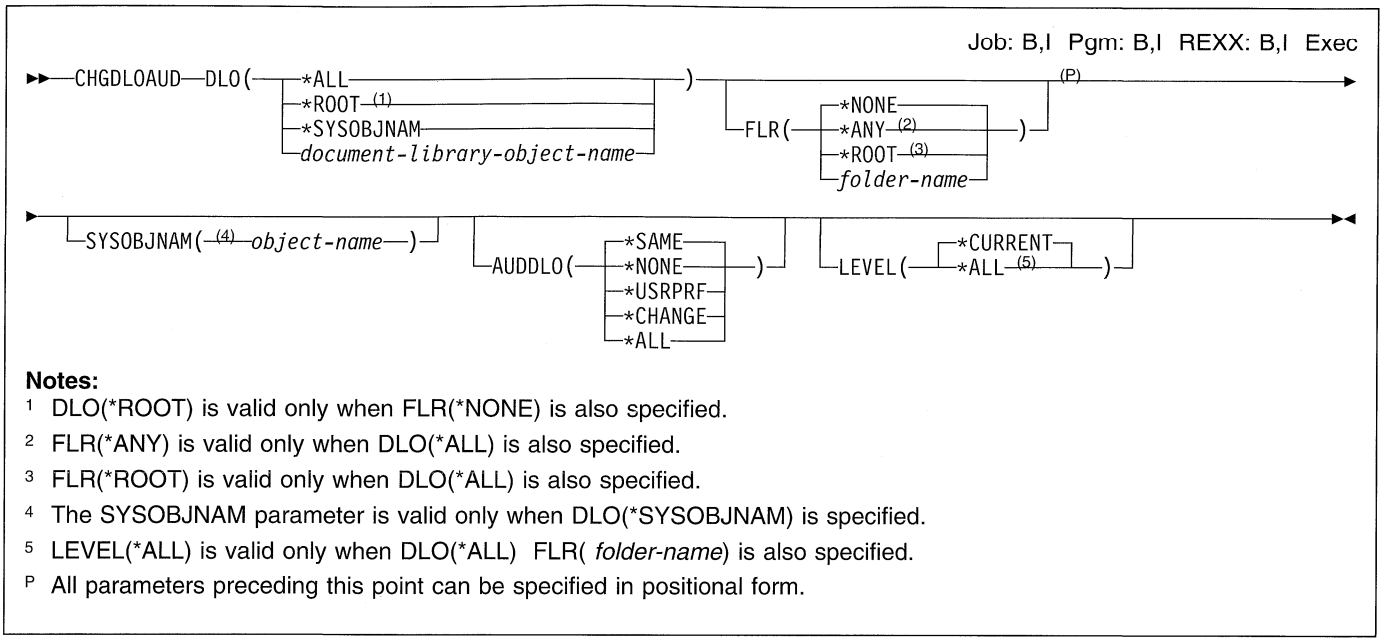
This command changes the diskette device file named PRNTRPT stored in the ACCREC library. The device file causes all input/output operations between the program and the diskette to be direct (without spooling). Other values in the file description do not change.

Example 2: Specifying DBCS Data Processing

```
CHGDKTF  FILE(IGCLIB/IGCDKT)  IGCDTA(*YES)
```

This command changes the diskette device file named IGCDKT, which is stored in the library IGCLIB, so that it processes double-byte character set data.

CHGDLOAUD (Change Document Library Object Audit) Command



Notes:

1 DLO(*ROOT) is valid only when FLR(*NONE) is also specified.

2 FLR(*ANY) is valid only when DLO(*ALL) is also specified.

3 FLR(*ROOT) is valid only when DLO(*ALL) is also specified.

4 The SYSOBJNAM parameter is valid only when DLO(*SYSOBJNAM) is specified.

5 LEVEL(*ALL) is valid only when DLO(*ALL) FLR(*folder-name*) is also specified.

P All parameters preceding this point can be specified in positional form.

Purpose

The Change Document Library Object Audit (CHGDLOAUD) command allows a user with *AUDIT special authority to change the auditing level of a document library object (DLO) or group of DLOs.

Notes:

1. Only the document library objects that exist when the command begins processing are changed. Objects created while the command is running might not be included in the change.
2. When changing objects in a folder (including the *ROOT level folder), the auditing level of the folder is changed first so any subsequent objects created into the folder inherit the containing folder's auditing level.
3. The *ROOT level folder is changed first when changing all objects on the system.
4. The default auditing level for newly created documents not contained in any folder is the value found in the QCRTOBJAUD system value. This system value should be changed to the desired auditing level prior to running the CHGDLOAUD command for documents not contained in any folder in order to ensure that all newly created documents not contained in any folder will inherit the proper auditing level.

Required Parameter

DLO

Specifies the name of the document or folder whose auditing level is changed.

***ALL:** The auditing level for all DLOs in the specified folder is changed.

***ROOT:** The auditing level for the *ROOT level folder is changed. The value specified on the AUDDLO parameter becomes the default auditing level for all new first-level folders.

***SYSOBJNAM:** The auditing level for the document or folder with the system object name specified on the SYSOBJNAM parameter is changed.

document-library-object-name: Specify the document library object for which the auditing level is changed.

Optional Parameters

FLR

Specifies the folder containing the document library object whose auditing level is changed.

***NONE:** The documents or folders that are not in a folder are changed.

***ANY:** All document library objects are changed.

***ROOT:** All first-level folders are changed. The *ROOT level folder is also changed.

folder-name: Specify the name of the folder that contains the document library object whose auditing level is changed. The folder specified on the command is also changed if DLO(*ALL) is specified.

AUDDLO

Specifies the auditing level for the document or folder changed.

***SAME:** The level does not change.

***NONE:** No read or change auditing occurs for the document or folder.

***USRPRF:** The current user profile is used to determine whether an audit record is sent for this session.

***CHANGE:** All change access to this DLO is logged.

***ALL:** All change and read access to this DLO is logged.

SYSOBJNAM
Specifies the system object name. This parameter is valid only when DLO(*SYSOBJNAM) is specified. Ten characters must be specified.

LEVEL
Specifies whether documents and folders at nested levels in the specified folder are changed.

***CURRENT:** Only the documents and folders at the current level are included in the change.

***ALL:** The documents and folders at all levels are included in the change.

Examples

Example 1: Changing an Auditing Level

```
CHGDLOAUD   DLO(MYDOC)  FLR(MYFLR)
  AUDDLO(*ALL)
```

This command changes the auditing level of document MYDOC in folder MYFLR so all change or read access to this DLO is logged in the security journal.

Example 2: Changing a System Object Name Auditing Level

```
CHGDLOAUD   DLO(*SYSOBJNAM)  SYSOBJNAM(FMCM210974)
  AUDDLO(*ALL)
```

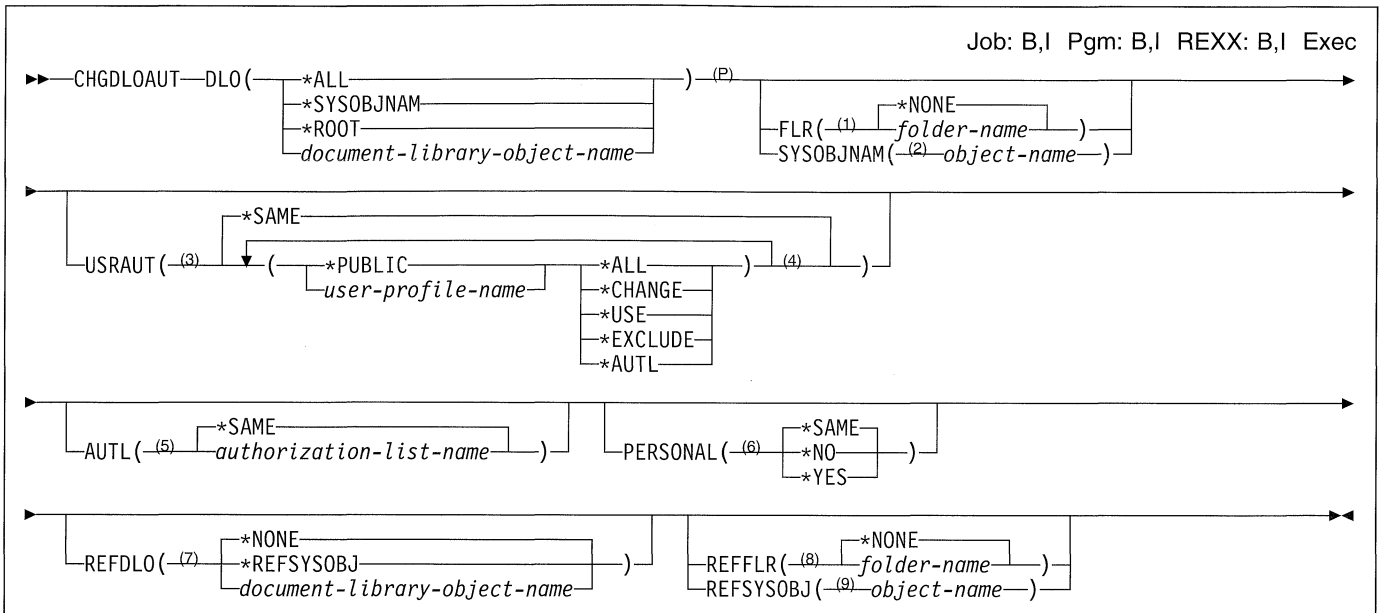
This command changes the auditing level of a document with the system object name FMCM210974 so all change or read access to this DLO is logged in the security journal.

Example 3: Changing the Auditing Level of all DLOs

```
CHGDLOAUD   DLO(*ALL)  FLR(*ANY)
  AUDDLO(*ALL)
```

This command changes the auditing level of all the DLOs so all change or read access to the DLOs is logged in the security journal.

CHGDLOAUT (Change Document Library Object Authority) Command



Notes:

1 FLR is not valid when DLO(*ROOT) is specified.

2 The SYSOBJNAM parameter is valid only when DLO(*SYSOBJNAM) is specified.

3 USRAUT(*PUBLIC *CHANGE) or USRAUT(*PUBLIC *USE) must be specified when DLO(*ROOT) is specified.

4 A maximum of 50 repetitions

5 AUTL is not valid when DLO(*ROOT) is specified.

6 PERSONAL is not valid when DLO(*ROOT) is specified.

7 REFDLO is not valid when DLO(*ROOT) is specified.

8 REFFLR is not valid when either *NONE or *REFSYSOBJ is specified for the REFDLO parameter.

9 REFSYSOBJ is only valid when *REFSYSOBJ is specified for the REFDLO parameter.

P All parameters preceding this point can be specified in positional form.

Purpose

The Change Document Library Object Authority (CHGDLOAUT) command is used to change an existing user's authority to a document or folder. This command can be used to:

- Change an existing user's specific authority
- Change the authorization list specifying the object's security
- Change public authority
- Change the existing personal status of the DLO

Restriction: The user of this command must have *ALL authority to the objects or *ALLOBJ special authority.

Required Parameter

DLO

Specifies the name of the document or folder whose user's authority is to be changed.

***ALL:** The user authority of all objects in the specified folder is changed. If *ALL is specified, the FLR parameter is required.

***SYSOBJNAM:** The user authority of the system object name specified in the SYSOBJNAM parameter is changed.

***ROOT:** The public authority value of the *ROOT folder is changed.

document-library-object-name: Specify the user-assigned name of the document or folder. Up to 12 characters can be specified.

Optional Parameters

FLR

Specifies the name of the folder that contains the document.

***NONE:** A folder name is not specified. If DLO(document-or-folder-name) is specified and the object is located in a folder, then FLR(*NONE) cannot be

specified. If DLO(*ALL) is specified, then FLR(*NONE) cannot be specified.

folder-name: Specify the user-assigned name of the folder. The folder name can consist of a series of folder names if the object specified in the DLO parameter is located in a folder that is contained in another folder. Up to 63 characters can be specified. The SYSOBJNAM parameter is ignored.

SYSOBJNAM

Specifies the system object name of the document or folder. Ten characters must be specified. The SYSOBJNAM parameter is valid only when DLO(*SYSOBJNAM) is specified.

USRAUT

Specifies the name of an existing user and the new user authority level.

When USRAUT((*PUBLIC *CHANGE)) is specified, all users can create first-level folders in the *ROOT folder. When USRAUT((*PUBLIC *USE)) is specified, only users with *ALLOBJ special authority can create first-level folders. Folder creation is the only function controlled by these values. Public authority is the only security value that can be specified for the *ROOT folder. Only *CHANGE and *USE public authorities can be specified for the *ROOT folder.

***SAME:** The value does not change.

Element 1: Users Affected By Authority Level Change

***PUBLIC:** The authority of users with no specific authority and who are not on the authorization list is changed.

user-profile-name: Specify the name of the user profile whose specific authority is to be changed.

Element 2: User Authority Levels

***ALL:** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence and specify the security for the object, change the object, and perform basic functions on the object. The user can change ownership of the document library object.

***CHANGE:** The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority.

***USE:** The user can perform basic operations on the document library object, such as display or print document or folder content and description information. The user is prevented from changing the object. Use authority provides object operational authority and read authority.

***EXCLUDE:** The user cannot access the document library object.

***AUTL:** The authority for *PUBLIC comes from the authorization list specified in the AUTL parameter. *AUTL is only valid for USER(*PUBLIC). If the security for the object is not currently specified with an authorization list, a message is returned.

AUTL

Specifies whether the existing authorization list is replaced by a different authorization list.

***SAME:** The value does not change.

authorization-list-name: Specify the name of the authorization list used.

PERSONAL

Specifies whether access is allowed when a user is working on behalf of another user who is authorized to the document or folder.

***SAME:** The value does not change.

***NO:** Access is allowed when a user is working on behalf of another.

***YES:** Access is not allowed when a user is working on behalf of another. PERSONAL(*YES) requires that USER(*PUBLIC) be *EXCLUDE. This value cannot be specified if the access code zero (0) is assigned to the object.

REFDLO

Specifies that all existing document or folder user authorities are replaced by user authorities of the referenced object including specific authorities, authority given to users with no specific authorities, authorization list authority, and access codes. Personal status does not change.

***NONE:** A referenced document library object is not specified.

***REFSYSOBJ:** The referenced object is identified by using a system object name specified in the REFSYS parameter.

document-library-object-name: Specify the name of the document or folder being referenced. Up to 12 characters can be specified.

REFFLR

Specifies the folder where the object specified in the REFDLO parameter is located. The REFFLR parameter and the REFSYSOBJ parameter are mutually exclusive.

***NONE:** A folder name is not specified.

folder-name: Specify the user-specified name of the referenced folder. The folder name can consist of a series of folder names if the object specified in the REFDLO parameter is located in a folder that is contained in another folder. Up to 63 characters can be specified.

REFSYSOBJ

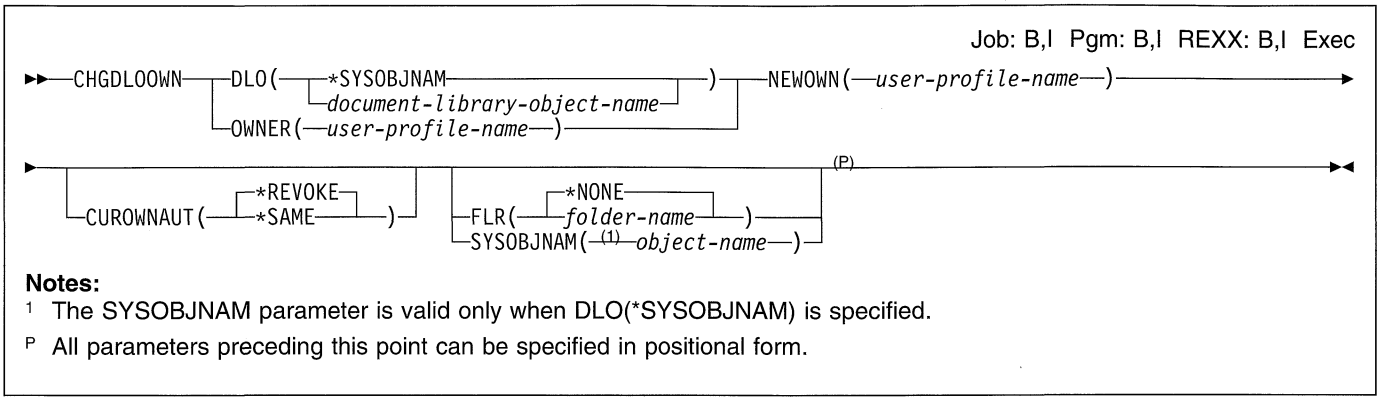
Specifies the system object name of the referenced object. A full 10 characters must be specified. The REFSYSOBJ parameter and the REFFLR parameter are mutually exclusive.

Example

```
CHGDLOAUT   DLO(MYDOC)  FLR(MYFLR)
USRAUT((*PUBLIC *AUTL))  AUTL(MYAUTL)
```

This command changes the authority of user *PUBLIC for document MYDOC in folder MYFLR. The authority specified on the authorization list for public (users with no specific authority for MYDOC, who are not on the authorization list MYAUTL, and whose user's group has no specific authority to MYDOC) is used to determine the public authority.

CHGDLOOWN (Change Document Library Object Owner) Command



Notes:
1 The SYSOBJNAM parameter is valid only when DLO(*SYSOBJNAM) is specified.
P All parameters preceding this point can be specified in positional form.

Purpose

The Change Document Library Object Owner (CHGDLOOWN) command transfers document or folder ownership from one user to another user; then the second user becomes the new owner. The new owner must be in the system distribution directory. The authorities that other users have to the document are not changed.

The owner of the object has *ALL authority for the object unless it is specifically changed. The owner has the authority to change any user's authority for the object. Owners can give any authorities to themselves that were specifically revoked previously. For example, owners can change their authority as a precaution, and then when they need the authority, they can give those authorities to themselves again.

Note: The system does not determine document or folder ownership by checking a group profile if one is specified in the user profile.

Restriction: To transfer ownership of a document or folder, the user must have *ALLOBJ authority or be the owner of the document or folder. The user must have delete authority to the old user profile and add authority to the new user profile.

Required Parameters

DLO
Specifies the name of the document or folder object that is assigned to a new owner. The DLO parameter and the OWNER parameter are mutually exclusive.

***SYSOBJNAM:** The object is identified using the system object name. Object name is specified by the SYSOBJNAM parameter.

document-library-object-name: Specify the name of the document or folder assigned by the user. Up to 12 characters can be specified.

OWNER
Specifies the user profile name of the current owner from whom all documents or folders are transferred. All documents and folders owned by this user are assigned a new owner. The OWNER parameter and the DLO parameter are mutually exclusive.

NEWOWN
Specifies the user profile name of the new owner to whom the documents or folders are transferred.

Optional Parameters

CUROWNAUT
Specifies whether the current owner's authority is revoked when ownership is transferred to the new owner specified on the NEWOWN parameter.

***REVOKE:** The current owner's authority is revoked when the object is transferred to the new owner.

***SAME:** The value does not change.

FLR
Specifies the name of the folder that contains the document.

***NONE:** No folder name is specified.

folder-name: Specify the user-assigned name of the folder. The folder name can consist of a series of folder names if the object specified in the OBJ parameter is located in a folder that is contained in another folder. Up to 63 characters can be specified.

SYSOBJNAM
Specifies the system object name. This parameter is valid only when DLO(*SYSOBJNAM) or DOCL(*SYSOBJNAM) is specified. A full ten characters must be specified.

Examples

Example 1: Transferring Folder Ownership

```
CHGDLOOWN   DLO(MYFLR)  NEWOWN(ANN)
```

This command assigns ownership of folder MYFLR to a new owner named ANN. The authority is revoked from the current owner.

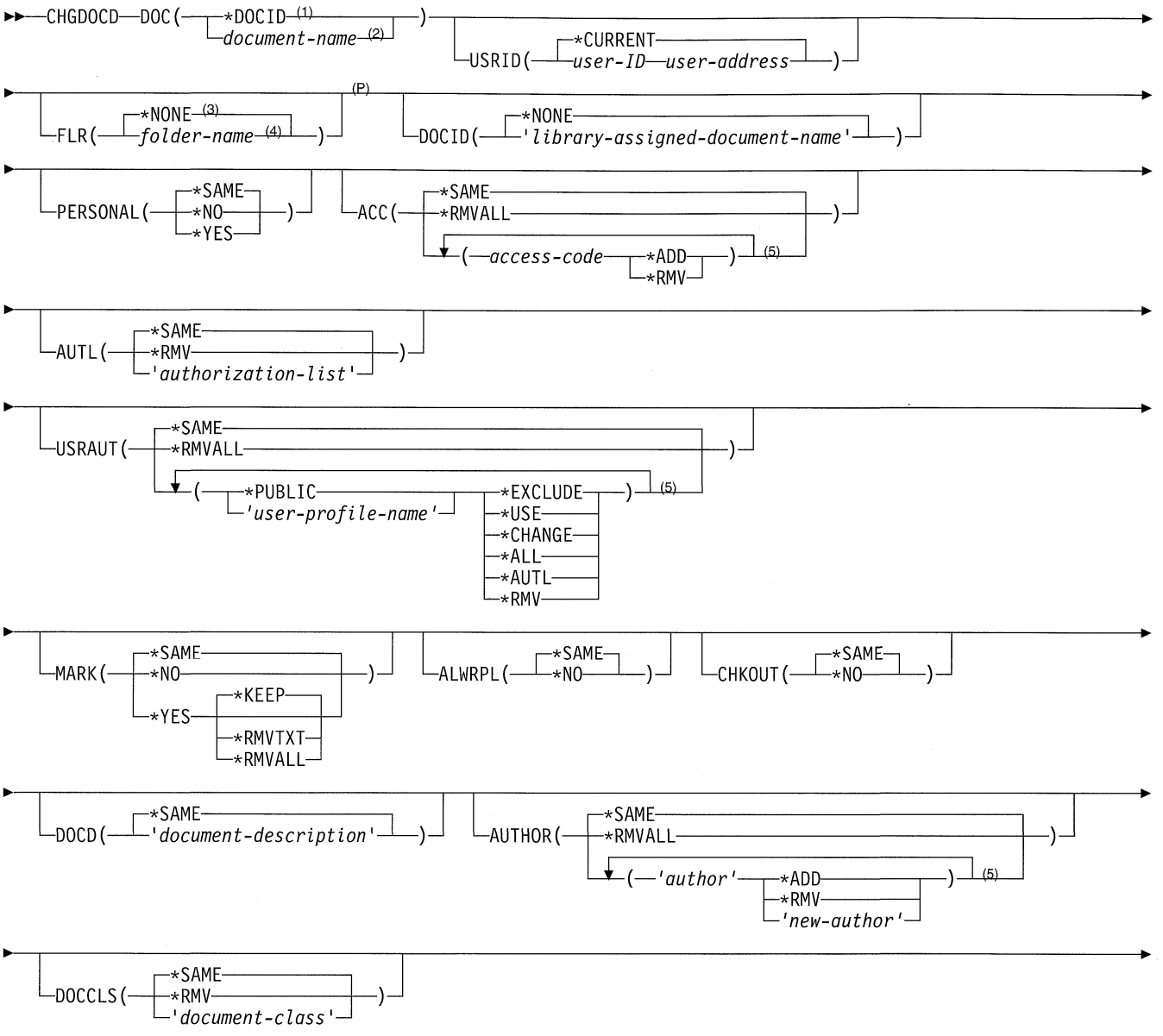
Example 2: Transferring Ownership of All Documents and Folders

```
CHGDLOOWN    OWNER(ANDERSON)   NEWOWN(SMITH)
```

This command assigns ownership of all documents and folders owned by ANDERSON to the new owner SMITH. The authority is revoked from the current owner.

CHGDOCD (Change Document Description) Command

Job: B,I Pgm: B,I REXX: B,I Exec



Notes:

1 DOCID(*library-assigned-document-name*) and FLR(*NONE) must be specified if this value is specified.

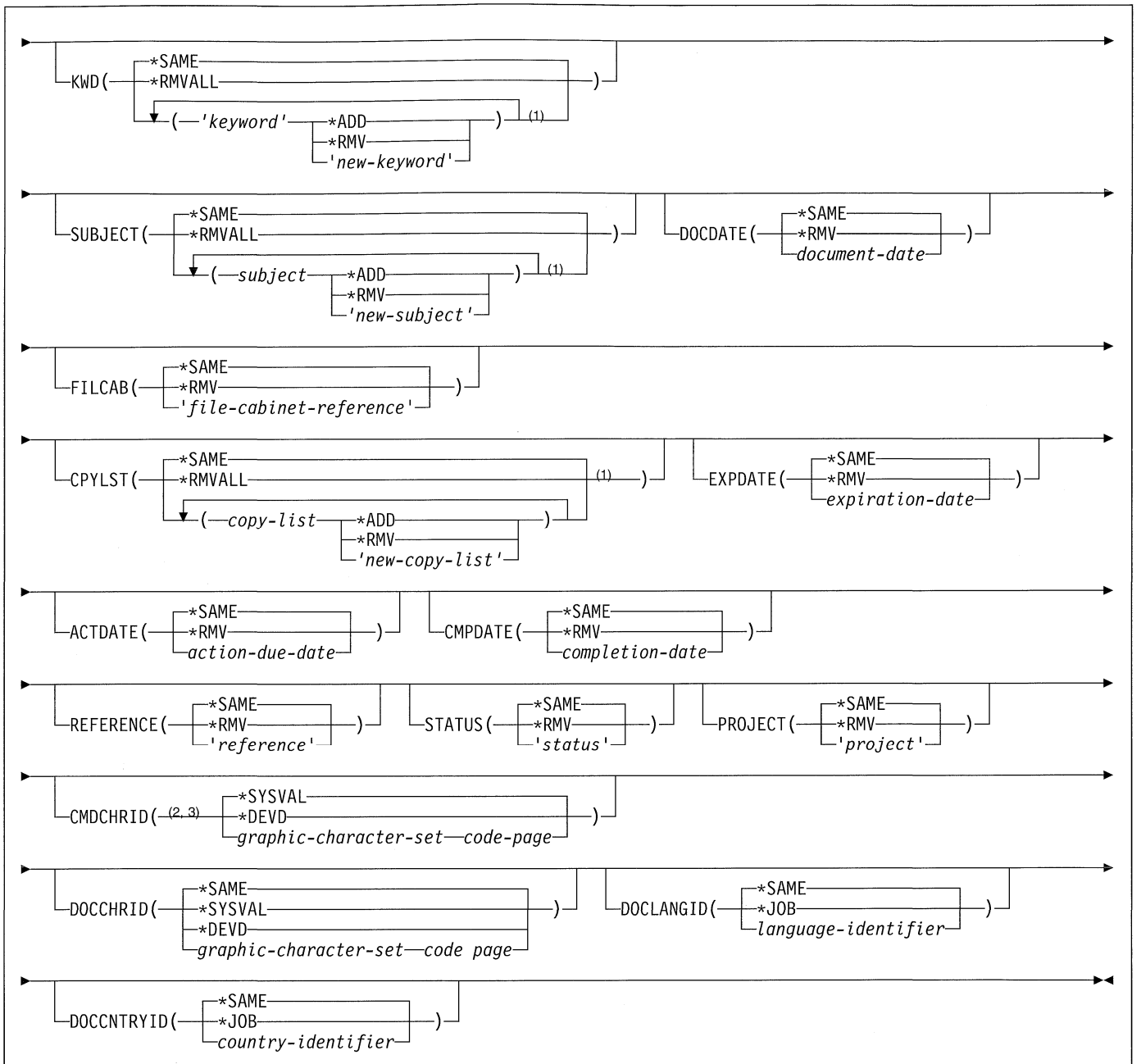
2 DOCID(*NONE) and FLR(*folder-name*) must be specified if this value is specified.

3 This value must be specified when DOC(*DOCID) is specified.

4 This value must be specified when DOC(*document-name*) is specified.

5 A maximum of 50 repetitions

P All parameters preceding this point can be specified in positional form.



Notes:

1 A maximum of 50 repetitions

2 This parameter applies to the first or only value of the DOCD, AUTHOR, USRID, DOCCLS, KWD, SUBJECT, FILCAB, CYPLST, REFERENCE, STATUS, and PROJECT parameters when the function is to replace or add the value.

3 This parameter applies to the second value of the AUTHOR, KWD, SUBJECT, and CPYLST parameters when the function is to add or replace the value.

Purpose

The Change Document Description (CHGDOCD) command allows the user to change descriptive information about a document that is already filed in the document library.

Restrictions:

1. To change the document profile values, you must have at least *CHANGE authority to the document, *ALLOBJ authority, or be working on behalf of a user who is authorized to the document.
2. To change the security values (for example, ACC, USRAUT, PERSONAL, AUTL), the checkout flag, and the allow replacement flag, you must be the owner of the document, have *ALL or *ALLOBJ authority, or be

working on behalf of a user who is authorized to the document.

3. Authority to work on behalf of others is granted with the Grant User Permission (GRTUSRPMN) command.

Required Parameter

DOC

Specifies the name of the document for which descriptive information is being changed on the local system.

***DOCID:** The document being changed is identified by the library-assigned document name specified on the DOCID parameter.

document-name: Specify the name of the document that is changed.

Optional Parameters

USRID

Specifies the user ID and address of the user for whom the document description is changed. The current user of this command must have the authority to work on behalf of the specified user profile. Authority to work on behalf of another user is granted with the GRTUSRPMN command.

***CURRENT:** The user profile under which the current job is running is used.

Element 1: User ID

user-ID: Specify the user ID of the user for whom the document description is changed.

Element 2: User Address

user-address: Specify the user address of the user for whom the document description is changed.

FLR

Specifies the name of the folder that contains the document.

***NONE:** No folder is specified. This value is required when DOC(*DOCID) is specified.

folder-name: Specify the user-assigned name of the folder that contains the document changed. The folder name can consist of a series of folder names (FLR1/FLR2/ and so forth) if the document specified in the DOC parameter is located in a folder that is contained in another folder. Up to 63 characters can be specified. This value is required when DOC(*document-name*) is specified.

DOCID

Specifies the library-assigned name of the document being changed. This is the name assigned to the document by the system when it was created. Documents filed outside the local system have only library-assigned document names. The library-assigned document names can be determined by using the Query Document Library (QRYDOCLIB) command or by the message returned from the File Document (FILDOC) command.

Library-assigned document names are 24 characters in length with the following format: YYYYMMDDHHMNSSHSSNSNSNSN where :

- YYYY = year
- MM = month
- DD = day
- HH = hour
- MN = minute
- SS = second
- HS = hundredths of a second
- SNSNSNSN = system name

***NONE:** There is no library-assigned document name. It is not required when the document is identified by the DOC parameter.

'library-assigned-document-name': Specify the name of the library assigned to the document being changed.

PERSONAL

Specifies whether the document being changed is a personal document. If it is, only the owner or an authorized user can access it. If the user entering this command is working on behalf of another user, that user must be authorized to the document.

***SAME:** The value does not change.

***NO:** The document is not a personal document.

***YES:** The document is a personal document.

ACC

Specifies the access codes that are added or deleted. Access codes are specified to allow groups of users access to the document. Access codes must be added on the system with the Add Access Code (ADDACC) command.

***SAME:** The value does not change.

***RMVALL:** All access codes are removed from the filed document.

Element 1: Access Code

access-code: Specify the access codes, ranging from 1 through 2047, that are added or deleted. The access code specified must exist on the system.

Element 2: Actions Permitted on Access Code

***ADD:** The specified access code is added to the access codes for the filed document.

***RMV:** The specified access code is removed from the document.

AUTL

Specifies that the authority for the document named in the DOC parameter comes from the current authorization list.

***SAME:** The value does not change.

***RMV:** The current authorization list is removed from the document.

'authorization-list': Specify the name of the authorization list whose authority replaces that of the current authorization list or adds an authorization list if one does not already exist.

USRAUT

Specifies the users that can access the document and the authority of each user. This parameter is used to change the authorization list to this document by removing or changing the user's authority to the document.

***SAME:** The value does not change.

***RMVALL:** All users are removed from the authorization list for the filed document.

Element 1: Users Affected by Authority Level Change

***PUBLIC:** The specific authority given to users who do not have specific authority to the document, who are not on the authorization list, and whose user's group does not have any specific authority to the document are changed.

'user-profile-name': Specify the user ID of a user added to or deleted from the specific user access list.

Element 2: User Authority Levels

***EXCLUDE:** The user cannot access the document description.

***USE:** The user can perform basic operations on the document description, such as running a program or reading a file. The user cannot change the document description. *USE authority provides object operational authority and read authority.

***CHANGE:** The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority.

***ALL:** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence and specify the security for the object, change the object, and perform basic functions on the object. The user can change ownership of the document description.

***AUTL:** The authority of the authorization list specified in the AUTL parameter is used for the document. AUTL is valid only if *PUBLIC is also specified.

***RMV:** The user is removed from the document.

MARK

Specifies whether the document is saved on media or in an online save file. Mark is a function of the Save Document Library Object (SAVDLO) command where documents that have been marked for safe, permanent storage (MARK(*YES)) are saved on disk, diskette, or tape.

This parameter can be specified as a single value or as a list of values (elements).

***SAME:** The value does not change.

***NO:** The mark for storage is turned off and the document is not saved by the SAVDLO command.

Element 1: Document Marked for Storage

***YES:** The mark for storage is turned on, and the document is saved by the SAVDLO command.

Element 2: How Document is Stored

***KEEP:** When the document is saved to disk, diskette, or tape, the document is also kept for online use.

***RMVTXT:** When the document is saved to disk, diskette, or tape, the document content is removed online, but the document profile is kept online for document library searches.

***RMVALL:** When the document is saved to disk, diskette, or tape, the document content and profile is removed online. No trace of the document is left online for the document library searches.

ALWRPL

Specifies the setting to allow replacement of the document content. Once a document has been changed so that it cannot be replaced, it cannot be changed back to a document that can be replaced. The current user of this command must have *ALL authority to the document to request this change.

***SAME:** The value does not change.

***NO:** The document content cannot be changed after this request is complete.

CHKOUT

Specifies the setting for the checkout flag value. With this parameter, users can reset the checkout status of a document already checked out by a retrieve request (RTVDOC command). A document cannot be checked out for update with this parameter. The CHKOUT parameter is ignored for documents that cannot be replaced. Only the document owner or users with *ALL authority can request the CHKOUT value to be set.

***SAME:** The value does not change.

***NO:** The checkout flag is set to off.

DOCD

Specifies the description of the document being changed. This is the Document Interchange Architecture Profile document name field.

***SAME:** The value does not change.

'document-description': Specify the current description of the document. Up to 44 characters can be specified.

AUTHOR

Specifies the name of the author of the document.

***SAME:** The value does not change.

***RMVALL:** All author names are removed.

Element 1: Author

'author': Specify an existing author name that is being changed or deleted, or a new author name that is being added.

Element 2: Actions Permitted for Author

***ADD:** The author name specified on the first element of this parameter is added as an author for the filed document.

***RMV:** The author name specified on the first element of this parameter is removed from the document.

'new-author': Specify the name of the new author to replace that of the author specified on the first element of this parameter. Up to 50 authors can be specified and each author name can have up to 20 characters.

DOCCLS

Specifies the class name associated with this document, such as MEMO, FORM, or SHEET.

***SAME:** The value does not change.

***RMV:** The document class is removed from the document.

'document-class': Specify the name of the document class to replace the existing document class or the document class to be added. Up to 16 characters can be specified.

KWD

Specifies the keywords that can be used to describe the document.

***SAME:** The value does not change.

***RMVALL:** All keywords are removed for the filed document.

Element 1: Keyword

'keyword': Specify a current keyword that is being removed or a new keyword that is being added.

Element 2: Actions Permitted for Keyword

***ADD:** The keyword is added for the filed document.

***RMV:** The keyword is removed from the document.

'new-keyword': Specify a new keyword to replace the current keyword. Up to 60 characters can be specified.

SUBJECT

Specifies the subject of the document.

***SAME:** The value does not change.

***RMVALL:** All subjects are removed for the filed document.

Element 1: Subject

'subject': Specify a current subject being removed or a subject being added.

Element 2: Actions Permitted for Subject

***ADD:** The subject is added for the filed document.

***RMV:** The subject is removed from the document.

'new-subject': Specify a new subject to replace the current subject. Up to 60 characters can be specified.

DOCDATE

Specifies any date the user wants to assign to the document.

***SAME:** The value does not change.

***RMV:** The document date is removed from the document.

document-date: Specify the document date being added or replaced. The date must be in the format specified by the system value QDATFMT.

FILCAB

Specifies the file cabinet location where the printed document is stored. This parameter is intended for printed documents. All that changes is the Document Interchange Architecture Profile field that refers to the printed document.

***SAME:** The value does not change.

***RMV:** The filing cabinet reference is removed from the document.

'file-cabinet-reference': Specify the file cabinet reference being added or replaced. Up to 60 characters can be specified.

CPYLST

Specifies the names of the users (in copy list entries) who receive this document.

***SAME:** The value does not change.

***RMVALL:** All copy list entries are removed from the filed document.

Element 1: Copy List

'copy-list': Specify a current copy list entry (the name of a recipient) that is changed or deleted or a new copy list entry being added.

Element 2: Actions Permitted on Copy List

***ADD:** The copy list entry is added to the filed document.

***RMV:** The copy list entry is removed from the document.

'new-copy-list': Specify up to 60 characters for the new copy list.

EXPDATE

Specifies the date on which the document is no longer needed.

***SAME:** The value does not change.

***RMV:** The expiration date is removed from the document.

expiration-date: Specify the date when the document is no longer needed.

ACTDATE
Specifies the date when the action requested is due.

***SAME:** The value does not change.

***RMV:** The action due date is removed from the document.

action-due-date: Specify the action due date being replaced or added. The date must be specified in the format specified by the system value QDATFMT.

CMPDATE
Specifies the date when the action requested is completed.

***SAME:** The value does not change.

***RMV:** The completion date is removed from the document.

completion-date: Specify the completion date being replaced or added. The date must be specified in the format specified by the system value QDATFMT.

REFERENCE
Specifies a reference associated with the document.

***SAME:** The value does not change.

***RMV:** The document reference is removed from the document.

'reference': Specify the document reference being replaced or added. Up to 60 characters can be used.

STATUS
Specifies the user-defined status (In Process, Pending Approval, or Retired).

***SAME:** The value does not change.

***RMV:** The document status is removed from the document.

'status': Specify the document status being replaced or added. Up to 20 characters can be specified.

PROJECT
Specifies the project with which the document is associated.

***SAME:** The value does not change.

***RMV:** The document project is removed from the document.

'project': Specify the document project being replaced or added. Up to 10 characters can be specified.

CMDCHRID
Specifies the character identifier (graphic character set and code page) for data being specified as parameter values on this command. This character identifier (CHRID) is related to the display device used to specify the command. More information about CHRID processing is in the *Guide to Programming Displays.*

Note: The CMDCHRID parameter applies to the following parameters and it means that the character set and code page are stored with the fields to allow the display station that accesses the document to correctly print or display the fields. The fields are translated to a common character set and code page when the fields are written to the search database. The interchangeable character set and code page set is '697 500'.

- DOCD
- AUTHOR
- DOCCLS
- KWD
- SUBJECT
- FILCAB
- CPYLST
- REF
- STATUS
- PROJECT

This value translates the USRID parameter to character set and code page of '930 500'. The *Communications: Distribution Services Network Guide* contains the character set and code page table for '930 500'.

***SYSVAL:** The system determines the graphic character set and code page values for the command parameters from the QCHRID system values.

***DEVD:** The system determines the graphic character set and code page values for the command parameter from the display device description where the command is entered. This option is valid only when specified from an interactive job. If this value is specified in an interactive CL program or a batch job, an error message is sent.

Element 1: Character Set

graphic-character-set: Specify the graphic character set values used to create the command parameter.

Element 2: Code Page

code-page: Specify the code page. Valid values range from 1 through 9999.

DOCCHRID
Specifies the character identifier (graphic character set and code page) being used for the document data. The character identifier is related to the display device used to create the document data.

***SAME:** The character identifier does not change.

***SYSVAL:** The graphic character set and code page values are determined from the QCHRID system value.

***DEVD:** The graphic character set and code page values defined for the device description where this

command is entered are used. This value cannot be specified if the command is run in batch.

Element 1: Character Set

graphic-character-set: Specify a maximum of 3 characters for the graphic character set.

Element 2: Code Page

code-page: Specify a maximum of 3 characters for the code page.

DOCLANGID

Specifies the language identifier being placed in this document's interchange document profile (IDP). This parameter can be used to add a language identifier to the document's IDP if the identifier has not been specified previously, or to change the value of an existing language identifier. If a value is specified on this parameter and no value has previously been specified on the DOCCNTRYID parameter, a country identifier must also be specified.

***SAME:** The language identifier does not change.

***JOB:** The language identifier specified for the job in which this command is entered is used.

language-identifier: Specify the language identifier.

DOCCNTRYID

Specifies the country identifier being placed in this document's interchange document profile (IDP). This parameter can be used to add a country identifier to the document's IDP if an identifier has not been specified previously, or to change the value of an existing country identifier. If a value is specified on this parameter and no value has previously been specified on the DOCLANGID parameter, a language identifier must also be specified.

***SAME:** The country identifier does not change.

***JOB:** The country identifier specified for the job in which this command is entered is used.

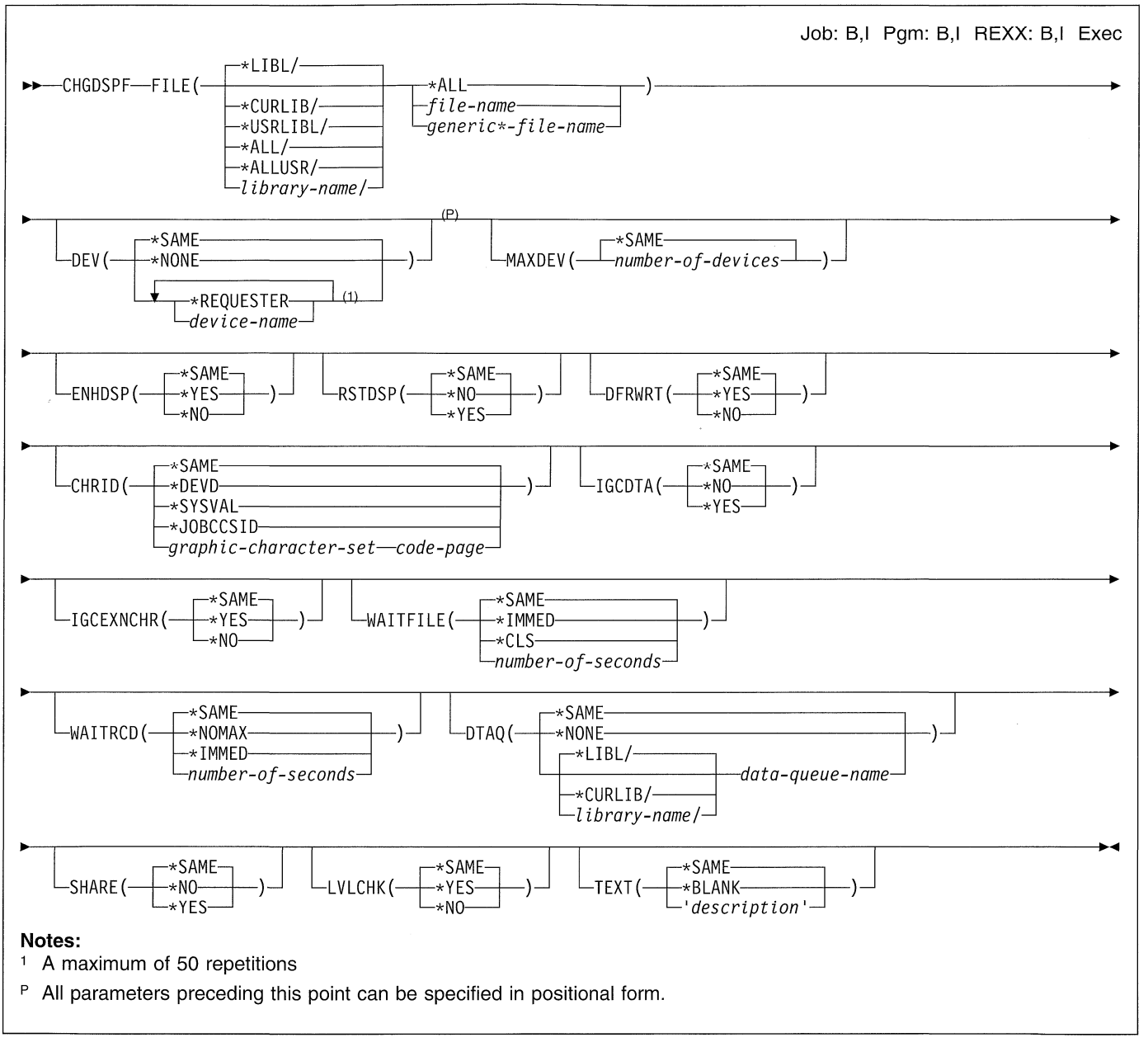
country-identifier: Specify the country identifier.

Example

```
CHGDOCD  DOC(DOC1) FLR(FLR1) PERSONAL(*YES)
  AUTL(USERAUTL) ALWRPL(*NO)  AUTHOR('AUTHOR1' *RMV)
```

This command changes document DOC1, which is located in folder FLR1, to a personal document. This document is no longer a document that can be replaced. AUTHOR1 is removed from the list of authors. If an authorization list already exists, it is replaced with authorization list USERAUTL; otherwise, USERAUTL is added as the authorization list for the document. The remaining information associated with the document does not change.

CHGDSPF (Change Display File) Command



Purpose

The Change Display File (CHGDSPF) command changes, in the file description, one or more of the attributes of the specified display device file.

Required Parameter

FILE

Specifies the qualified name of the display device file whose description is being changed.

Depending on the library qualifier specified or assumed, the following libraries, for which the user has the authority, are searched for the specified objects:

The name of the display device file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

***USRLIBL:** Only the libraries in the user portion of the job's library list are searched.

***ALL:** All libraries in the system portion of the job's library list, including QSYS, are searched.

***ALLUSR:** All user libraries are searched. All libraries with names that do not begin with the letter Q are searched except for the following:

```
#CGULIB     #DFULIB     #RPGLIB     #SEULIB
#COBLIB     #DSULIB     #SDALIB
```

Although the following Qxxx libraries are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are considered *user libraries*, and are also searched:

```
QDSNX       QPFRDATA    QUSER38
QGPL        QRCL        QUSRSYS
QGPL38      QS36F       QUSRVxRxMx
```

Note: A different library name, of the form QUSRVxRxMx, is added with each release. VxRxMx is the version, release, and modification level of the library.

library-name: Specify the name of the library to be searched.

***ALL:** All the display device files in the specified library are changed.

file-name: Specify the file name that is to be changed.

generic-file-name:* Specify the generic name of the file. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be changed only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

Optional Parameters

DEV

Specifies the names of one or more display devices used with this display device file to pass data records between the users of the display devices and their jobs. The device name specified in the display device file supplied by IBM is *REQUESTER.

***SAME:** The value does not change.

***NONE:** No device name is specified. The name of the display device must be specified later in a CHGDSPF or OVRDSPF command, or in the HLL program that opens the file.

***REQUESTER:** The device from which the program is called is assigned to the file when the file is opened.

device-name: Specify the names of one or more display devices used with this device file to pass data records between the users of the devices and the system. Each device name must already be known on the system by a device description before this device file is created. *REQUESTER can be specified as one of the names. Up to 50 names can be specified in this command, but the total number cannot exceed the number specified on the MAXDEV parameter.

MAXDEV

Specifies the maximum number of display devices that are connected to the display device file at the same time, while the file is open. However, if a CL program is written to get access to more than one work station through the same file (through a single running of the program), this parameter must specify a value greater than 1.

The names of the devices are specified in the DEV parameter of this command, in a later CHGDSPF or OVRDSPF command, or in the HLL program that opens the file.

***SAME:** The value does not change.

number-of-devices: Specify the maximum number of devices that are connected to this display file at the same time. Valid values range from 1 through 256.

ENHDSP

Specifies whether the data being shown at a display station by this display file is using the enhanced capabilities available on the display station.

***SAME:** The value does not change.

***YES:** The data for the display file is shown using any enhanced capabilities available on the display station. These capabilities can include mnemonics, selection cursor, and graphical window borders.

***NO:** The data for this display file is shown as it would be on a 5250 display station. No enhanced capabilities that are available on the display, such as mnemonics, selection cursor, or graphical window borders, are used. This value is normally used to preserve character-based interaction across all display stations.

RSTDSP

Specifies whether data being shown on a display by this display file is saved at the time the file is suspended (made temporarily inactive) so that another display file can show different data on the same device. If the data for this file is saved, it is restored to the display of the device when the file is used again.

***SAME:** The value does not change.

***NO:** The data being shown by this file is not saved when the file is suspended. When control is returned to the programs using this file, the data is not restored.

***YES:** The data being shown when the file is suspended is saved so it can be shown on the display when the file is used again.

DFRWRT

Specifies that the writing of data is deferred (delayed) until it is written out with other data when a read request is made. Control is returned to the program immediately

after the data is received. This may result in improved performance.

***SAME:** The value does not change.

***YES:** When the program issues a write request, control is returned after the buffer is processed. The data may not be shown immediately; the actual display of the data may take place later when a read or combined write/read operation is performed. The buffer is then available to be prepared for the next read or combined write/read operation.

***NO:** After a write operation, the user program does not regain control until the input/output operation is completed (with the data displayed and the input/output feedback information available).

CHRID

Specifies the character identifier (graphic character set and code page) that a work station display device supports. When a display file that was created with the CHRID DDS keyword is used with the device, the system converts data sent to and received from the device to ensure that the correct characters are shown and that the correct hexadecimal byte values are returned to the application program. More information about display file CHRID processing and the translation tables that are used to convert data sent to and received from the display are in the *Guide to Programming Displays*.

***SAME:** The value does not change.

***DEVD:** The value specified on the CHRID parameter in the device description of the work station on which the application is running, is used. If no CHRID value is specified, the QCHRID system value for the system on which the application is running, is used. No conversion is necessary because the file has the same character identifier as the work station. For a list of valid values, see Table 30 in the CRTDEVDSP command description.

***SYSVAL:** The system determines the graphic character set and code page values for the command parameters from the QCHRID system values.

| ***JOBCCSID:** The character data is changed from the device CHRID to the CCSID (coded character set identifier) of the job on display file input, and from the CCSID of the job to the device CHRID on display file output.

| **Note:** This value is not allowed if the file was created on a system at an earlier release level than V2R3M0.

Element 1: Character Set

graphic-character-set: Specify the graphic character set values that match the attributes of the display device. Valid values range from 1 through 32767.

Element 2: Code Page

code-page: Specify the code page set values that match the attributes of the display device. Valid values range from 1 through 32767.

IGCDTA

Specifies, for program-described original files, whether the file processes double-byte character set (DBCS) data. For externally described printer files, this parameter specifies DBCS attributes of the file.

The possible values for program-described files are:

***SAME:** The value does not change.

| ***NO:** The file does not process DBCS data.

| ***YES:** The file processes DBCS data.

IGCEXNCHR

Specifies whether the system processes double-byte character set (DBCS) extension characters.

***SAME:** The value does not change.

***YES:** The system processes DBCS extension characters.

***NO:** The system does not process DBCS extension characters; it displays extension characters as the undefined character.

WAITFILE

Specifies the number of seconds that the program waits for the file resources and session resources to be allocated when the file is opened, or for the device or session resources to be allocated when an acquire operation is performed to the file. If those resources are not allocated within the specified wait time, an error message is sent to the program. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

Note: An immediate allocation of the device by the device resource is required when an acquire operation is performed to the file.

***SAME:** The value does not change.

***IMMED:** The program does not wait; when the file is opened, an immediate allocation of the file resources is required.

***CLS:** The job default wait time is used as the wait time for the file resources being allocated.

number-of-seconds: Specify the number of seconds that the program waits for the file resources to be allocated to the display device file when the file is opened, or the wait time for the device allocated when an acquire operation is performed to the file. Valid values range from 1 through 32767 seconds.

WAITRCD

Specifies the number of seconds the program waits for the completion of a read-from-invited-device operation to a multiple device file in a high-level language program. Refer to the appropriate high-level language reference manual to determine when a file is treated as a multiple device file. The program performing the read operation waits for input from all invited devices currently accessing the file. If a record is not returned from an invited device in the specified amount of time, a notify

message is sent to the program. This parameter has no effect on an input operation directed to a specific device.

Note: This parameter is also used to specify the time (seconds) that a CL program waits to complete a WAIT command. If a record is not returned from any of the devices that should return a record, an escape message is sent to the CL program. More information on the WAITRCD parameter is in the Receive File (RCVF), Send File (SNDF), Send/Receive File (SNDRCVF), and WAIT (Wait) command descriptions.

***SAME:** The value does not change.

***NOMAX:** There is no limit on the time the system waits for the completion of the operation.

***IMMED:** The program does not wait for the read-from-invited-device operation for the completion of the file. A record must be available from an invited program device when the read-from-invited-program-device operation is performed. If a record is not already available when the read-from-invited-program-device operation is performed, a notify message is sent to the program.

number-of-seconds: Specify the number of seconds that the program waits for the completion of the read-from-invited-device operation. Valid values range from 1 through 32767.

DTAQ

Specifies the name of the data queue that receives an entry from the system when a data-available event is signaled from an invited display device. The data queue need not exist when the display file is created since the name specified on this parameter is not evaluated until the file is used. More information on the data queue function is in the *CL Programmer's Guide*.

***SAME:** The value does not change.

***NONE:** A data queue does not receive an entry from the system.

The name of the data queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

data-queue-name: Specify the name of the data queue that is to receive an entry from the system when the data-available event is signaled.

SHARE

Specifies whether the open data path (ODP) for the display file is shared with other programs in the routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

More information on shared database files is in the *Database Guide*.

***SAME:** The value does not change.

***NO:** The ODP created by the program with this attribute is not shared with other programs in the routing step. Every time a program opens the file with this attribute, a new ODP to the file is created and activated.

***YES:** The ODP created with this attribute is shared with each program in the routing step that also specifies SHARE(*YES) when it opens the file.

Note: When SHARE(*YES) is specified and control is passed to a program, a read operation in that program retrieves the next input record. A write operation produces the next output record.

LVLCHK

Specifies whether the record format level identifiers in the program are checked against those in the device file when the file is opened. If so, the record format identifiers in the program must match those in the device file. Because the same record format name can exist in more than one file, each record format is given an internal system identifier when it is created.

***SAME:** The value does not change.

***YES:** The level identifiers of the record formats are checked when the file is opened. If the level identifiers do not match, an error message is sent to the program that requested the open, and the file is not opened.

***NO:** The level identifiers are not checked when the file is opened.

TEXT

Specifies text that briefly describes the display device file. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Examples

Example 1: Specifying Multiple Display Devices

```
CHGDSPF   FILE(ORDENT)  DEV(WS1 WS2 WS3)  MAXDEV(3)
```

This command changes the description of the display device file named ORDENT. The file is located through the library list. The devices used with this file are the work stations WS1, WS2, and WS3. All three of the devices can be used at the same time with this display file.

Example 2: Delaying Writing of Data

```
CHGDSPF   FILE(ACCREC/*ALL)  DFRWRT(*YES)
```

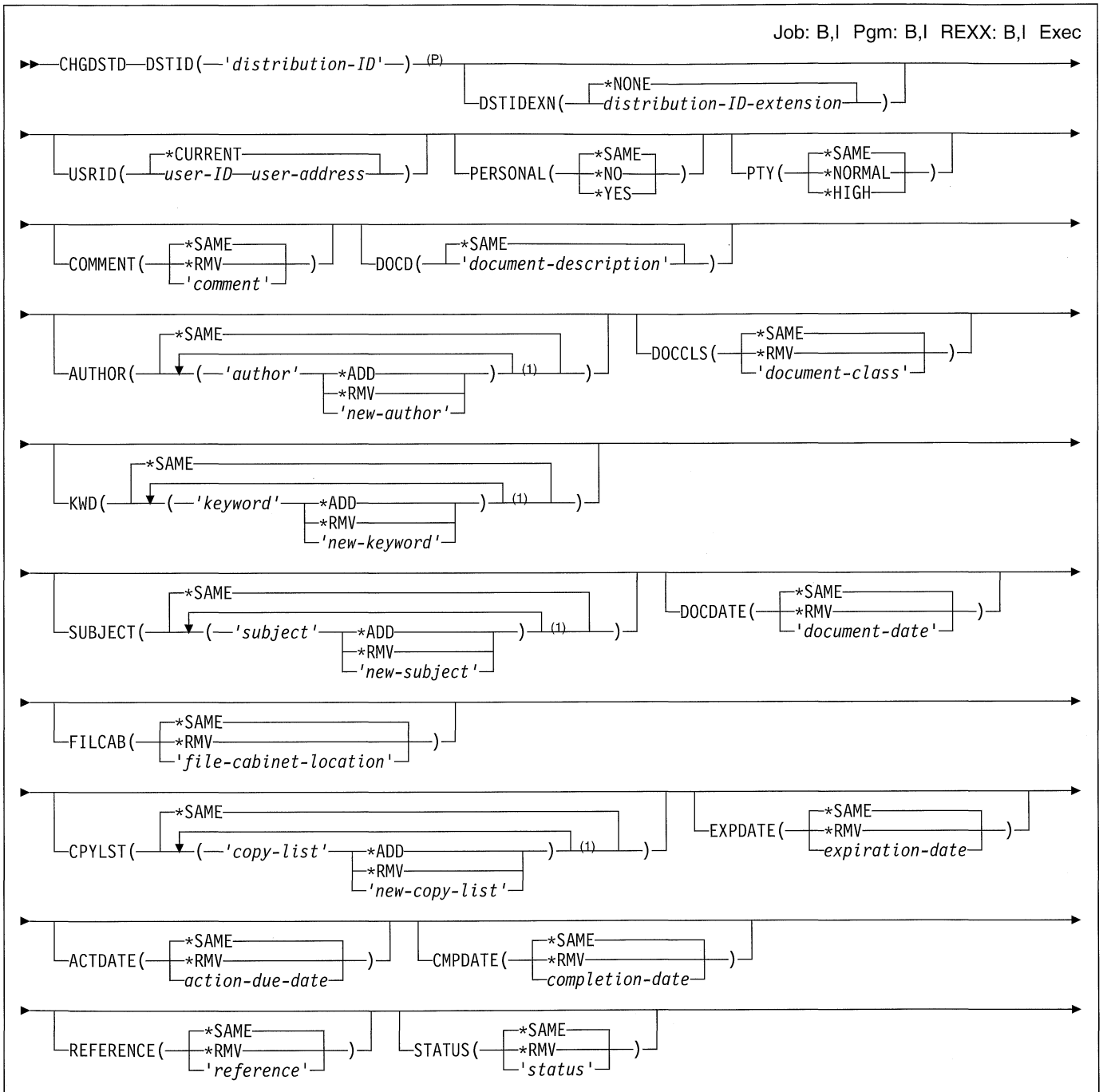
This command changes the description of all display files in library ACCREC to delay writing data until a read request is made.

Example 3: Specifying DBCS Data Processing

```
CHGDSPF   FILE(IGCLIB/IGCDSP) IGCDTA(*YES)
```

This command changes the display device file IGCDSP, which is stored in the library IGCLIB, so that it processes double-byte character set data.

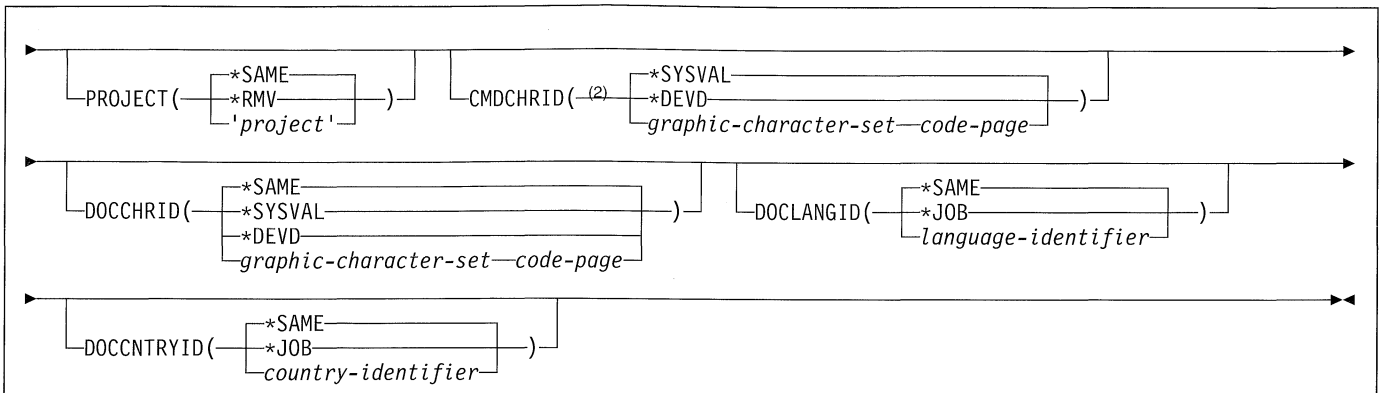
CHGDSTD (Change Distribution Description) Command



Notes:

1 A maximum of 50 repetitions

P All parameters preceding this point can be specified in positional form.



Notes:

1 A maximum of 50 repetitions

2 This parameter applies to USRID and DSTID parameters. When the function replaces the value, this parameter applies to the first and second value of the AUTHOR, KWD, SUBJECT, and CPYLST parameters. When the function adds a value, this parameter applies to the first value of the AUTHOR, KWD, SUBJECT, and CPYLST parameters. When the function adds or replaces a value, this parameter applies to the value of the DOCD, DOCCLS, FILCAB, REFERENCE, STATUS, and PROJECT parameters.

Purpose

The Change Distribution Description (CHGDSTD) command changes descriptive information or access information about a distribution document in the mail log. In addition, descriptive information can be deleted or added for some parameters.

When an existing distribution document interchange document profile parameter is changed and that parameter can have multiple occurrences, the current value must be supplied exactly as it was when the distribution document was created. For example, if one of the authors supplied with the distribution was *John L. Jones*, the AUTHOR parameter on this command must contain that exact character string if it is being changed. Spaces, special characters, and uppercase and lowercase characters are all significant. If an interchange document profile parameter (for example, the document name or class) allows only one occurrence, only the new value must be supplied to change the current value or add the parameter.

Restrictions: (1) A user can change the distribution description on behalf of another by specifying the other user's user ID and address in the USRID parameter. A user must be granted authority to work on behalf of another with the Grant User Permission (GRTUSRPMN) command. (2) The user ID and address must be enrolled in the system directory. (3) Personal distribution cannot be requested by a user who is working on behalf of another user.

Required Parameter

DSTID

Specifies the unique distribution identifier (ID) of the distribution. The ID is assigned to the distribution by the system that originated it.

Only incoming distributions can be changed. If the ID represents an outgoing distribution, an error message is returned.

'distribution-ID': Specify the sender's address (padded on the right to 8 characters), the sender's user ID (padded on the right to 8 characters), and a 4-digit zoned sequence number with leading zeros. For example, *NEWYORKbSMITHbbb0204*. The distribution ID is entered this way because a blank character is valid in a user ID address. The distribution ID can be determined by using the Query Distribution (QRYDST) command.

Optional Parameters

DSTIDEXN

Specifies the extension of the distribution ID. This extension, 2 digits that range from 01 through 99, identifies duplicate distributions. For example, if the distribution ID is *NEWYORKbSMITHbbb0204*. and two copies of this distribution are sent to a user, the user has two distributions with the same distribution ID. To uniquely distinguish the two distributions, an extension is added to each distribution. For example, one distribution is identified by distribution ID and extension *NEWYORKbSMITHbbb020401* and the other one by *NEWYORKbSMITHbbb020402*. If there are no duplicate IDs, the extension is automatically set to 01. These extensions map one to one with the distribution ID that is specified on the DSTID parameter.

***NONE:** There is no duplicate distribution. This value is equivalent to an extension of 01.

distribution-ID-extension: Specify the extension associated with the distribution. This is used to uniquely identify duplicate or multiple distributions.

USRID

Specifies the user ID and address of the user for whom this request is made.

***CURRENT:** The user profile under which the current job is running is used.

Element 1: User ID

user-ID: Specify the user ID of the user authorized to the distribution in which information is being changed.

Element 2: User Address

user-address: Specify the user address of the user authorized to the distribution in which information is being changed.

PERSONAL

Specifies whether the document distribution is a personal or nonpersonal.

***SAME:** The value does not change.

***NO:** Only the owner and persons that have authorization to the distribution document can get access to nonpersonal distributions. Users authorized to work on behalf of other users who have access to the distribution can access non-personal documents.

***YES:** Only the owner can get access to personal distribution documents. Users authorized to work on behalf of other users who have access to the distribution document cannot get access to the distribution.

PTY

Specifies the priority of the distribution document.

***SAME:** The value does not change.

***NORMAL:** The distribution document has normal priority.

***HIGH:** The distribution document has high priority.

COMMENT

Specifies user-defined comment text associated with the distribution of the document.

***SAME:** The value does not change.

***RMV:** The comment text is removed from the document.

'comment': Specify up to 256 characters of text, enclosed in apostrophes, associated with the distribution.

DOCD

Specifies the user-defined description of the distribution document being changed. This value occupies the Document Interchange Architecture Profile description name field.

***SAME:** The value does not change.

'document-description': Specify up to 44 characters of text, enclosed in apostrophes, that describes the distribution document.

AUTHOR

Specifies the names of the authors of the distribution document.

***SAME:** The value does not change.

Element 1: Author

'author': Specify the current author name being changed or removed, or specify the name of the author being added. Up to 20 characters can be specified, enclosed in apostrophes.

Element 2: Actions Permitted for Current Author

***ADD:** The specified author name is added to the information about the distribution document.

***RMV:** The specified author name is removed from the document.

'new-author': Specify the new author name that replaces the author name currently associated with the distribution document. Up to 20 characters can be specified, enclosed in apostrophes.

DOCCLS

Specifies the document class name associated with this distribution document.

***SAME:** The value does not change.

***RMV:** The document class name is removed from the document.

'document-class': Specify, enclosed in apostrophes, the document class being replaced or added. Valid values range from 1 through 16 characters.

KWD

Specifies the keywords that can be used to describe the distribution document.

***SAME:** The value does not change.

Element 1: Keyword

'keyword': Specify a current keyword being changed, removed, or a new keyword being added. Up to 60 characters can be specified, enclosed in apostrophes.

Element 2: Actions Permitted for Keyword

***ADD:** The specified keyword is added to the information about the document.

***RMV:** The specified keyword is removed from the document.

'new-keyword': Specify a new keyword to replace the current keyword. Up to 60 characters can be specified, enclosed in apostrophes.

SUBJECT

Specifies the subject of the distribution.

***SAME:** The value does not change.

Element 1: Subject

'subject': Specify the subject being added, changed, or removed. Up to 60 characters can be specified, enclosed in apostrophes.

Element 2: Actions Permitted for Subject

***ADD:** The subject of the distribution is added to the information about the distribution document.

***RMV:** The subject of the distribution is removed from the document.

'new-subject': Specify a new subject to replace the current subject. Up to 60 characters can be specified, enclosed in apostrophes.

DOCDATE

Specifies any date the user wants to assign to the document. The document date must be in the format specified by the system value QDATFMT.

***SAME:** The value does not change.

***RMV:** The document date is removed from the document.

'document-date': Specify the document date being replaced or added.

FILCAB

Specifies the physical location where the distribution is stored. This parameter is intended for printed distributions. The only change is in the Document Interchange Architecture Profile field that refers to the printed document.

***SAME:** The value does not change.

***RMV:** The file cabinet location is removed from the document.

'file-cabinet-location': Specify the file cabinet location being replaced or added. Up to 60 characters can be specified.

CPYLST

Specifies the names of the users (in copy list entries) who receive this distribution.

***SAME:** The value does not change.

Element 1: Copy List

'copy-list': Specify, enclosed in apostrophes, a current copy list entry (the name of a recipient) being changed, removed, or added.

Element 2: Actions Permitted for Copy List

***ADD:** The specified recipient name is added to the information about the distribution document.

***RMV:** The specified recipient name is removed from the document.

'new-copy-list': Specify, enclosed in apostrophes, the new copy list entry to replace the current copy list entry.

EXPDATE

Specifies the date on which the document is no longer needed.

Note: The expiration date must be formatted as specified by the system value QDATFMT.

***SAME:** The value does not change.

***RMV:** The expiration date is removed from the document.

expiration-date: Specify the expiration date being replaced or added.

ACTDATE

Specifies the due date for the requested action. The action due date must be in the format specified by the system value QDATFMT.

***SAME:** The value does not change.

***RMV:** The action due date is removed from the document.

action-due-date: Specify the action due date being replaced or added.

CMPDATE

Specifies the completion date for the requested action. The completion date must be in the format specified by the system value QDATFMT.

***SAME:** The value does not change.

***RMV:** The completion date is removed from the document.

completion-date: Specify the completion date being replaced or added.

REFERENCE

Specifies a reference associated with the distribution.

***SAME:** The value does not change.

***RMV:** The reference is removed from the document.

'reference': Specify the document reference that is replaced or added. Up to 60 characters can be specified, enclosed in apostrophes.

STATUS

Specifies the user-defined status (In Process, Pending Approval, or Retired).

***SAME:** The value does not change.

***RMV:** The status statement is removed from the document.

'status': Specify the document status that is replaced or added. Up to 20 characters can be specified, enclosed in apostrophes.

PROJECT

Specifies the name of the project with which the distribution document is associated.

***SAME:** The value does not change.

***RMV:** The project name is removed from the document.

'project': Specify the name of the document project that is replaced or added. Up to 10 characters can be specified, enclosed in apostrophes.

CMDCHRID

Specifies the character identifier (graphic character set and code page) for data being specified as parameter

values on this command. This character identifier (CHRID) is related to the display device used to specify the command. More information about CHRID processing is in the *Guide to Programming Displays*.

Note: The CMDCHRID parameter applies to the following parameters for which the data is translated to a common character set and code page for the search data base. The common character set and code page is 697 500, except for USRID and DSTID which is 930 500. The character set and code page are also attached to the field and are stored with the field to allow the terminal that gets access to the document to correctly print or display the field.

```
AUTHOR     DSTID
COMMENT    FILCAB
CPYLST     KWD
DOCCLS     SUBJECT
DOCD       USRID
```

***SYSVAL:** The system determines the graphic character set and code page values for the command parameters from the QCHRID system values.

***DEVD:** The system determines the graphic character set and code page values for the command parameter from the display device description where the command is entered. This option is valid only when specified from an interactive job. If this value is specified in an interactive CL program or a batch job, an error message is sent.

Element 1: Character Set

graphic-character-set: Specify the graphic character set values used to create the command parameters.

Element 2: Code Page

code-page: Specify the code page. Valid values range from 1 through 9999.

DOCCHRID

Specifies the character identifier (graphic character set and code page) being used for the document data. The character identifier is related to the display device used to create the document data.

***SAME:** The character identifier does not change.

***SYSVAL:** The graphic character set and code page values are determined from the QCHRID system value.

***DEVD:** The graphic character set and code page values defined for the device where this command is entered are used. This value cannot be specified if the command is run in batch.

Element 1: Character Set

graphic-character-set: Specify a maximum of 3 characters for the graphic character set.

Element 2: Code Page

code-page: Specify a maximum of 3 characters for the code page.

DOCLANGID

Specifies the language identifier being placed in this distribution document's interchange document profile (IDP). This parameter can be used to add a language identifier to the document's IDP if the identifier has not been specified previously, or to change the value of an existing language identifier. If a value is specified on this parameter and no value has previously been specified on the DOCCNTRYID parameter, a country identifier must also be specified.

***SAME:** The language identifier does not change.

***JOB:** The language identifier specified for the job in which this command is entered is used.

language-identifier: Specify the language identifier.

DOCCNTRYID

Specifies the country identifier to be placed in this distribution document's interchange document profile (IDP). This parameter can be used to add a country identifier to the document's IDP if an identifier has not been specified previously, or to change the value of an existing country identifier. If a value is specified on this parameter and no value has previously been specified on the DOCLANGID parameter, a language identifier must also be specified.

***SAME:** The country identifier does not change.

***JOB:** The country identifier specified for the job in which this command is entered is used.

country-identifier: Specify the country identifier.

Examples

Example 1: Deleting an Author

```
CHGDSTD  DSTID('NEWYORK SMITHbbb0201')
  DSTIDEXN(01)
  AUTHOR('John W. Baker' *DLT)
  KWD('PC Attached Scanners' *ADD)
```

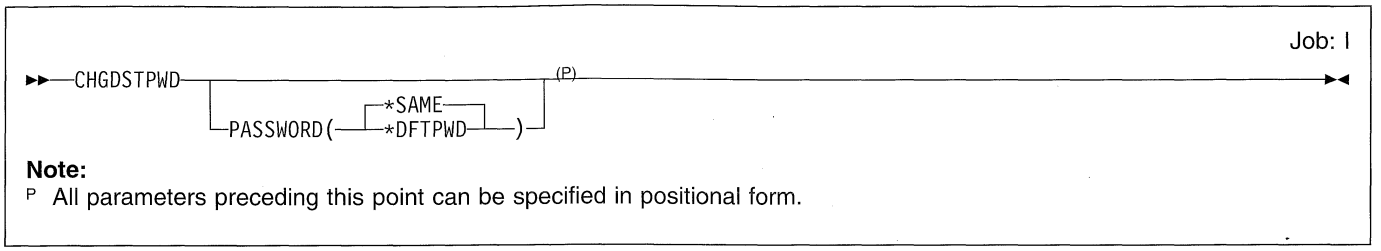
This command deletes one of the authors associated with a distribution document and adds a new keyword to the distribution document.

Example 2: Changing the User-Defined Document Name

```
CHGDSTD  DSTID('NEWYORK SMITHbbb0201')
DSTIDEXN(02)
  DOCD('S/38 1984 Sales')
  USRID(JACOBSON RCHJACOB)
```

In this example, a person authorized to work on behalf of JACOBSON is changing the user-defined document name associated with a document on the second distribution sent to the user.

CHGDSTPWD (Change Dedicated Service Tools Password) Command



Note:

P All parameters preceding this point can be specified in positional form.

Purpose

The Change Dedicated Service Tools Password (CHGDSTPWD) command is used to reset the dedicated service tools (DST) password to the system-shipped default. The DST password can be changed to the default only through this command. The DST password is changed immediately and a message is logged to the history file stating that it was changed.

Restrictions:

1. This command is shipped with public *USE authority, but you must be signed on as QSECOFR to use this command. You cannot authorize other users to the command.
2. Only the security officer (QSECOFR user profile) can reset the DST password to the system-shipped default.

Optional Parameter

PASSWORD

Specifies the value used to change the DST password.

***SAME:** The value does not change.

***DEFAULT:** The system-shipped default value is used to reset the DST password.

Examples

Example 1: Changing to the Default DST Password

```
CHGDSTPWD  PASSWORD(*DEFAULT)
```

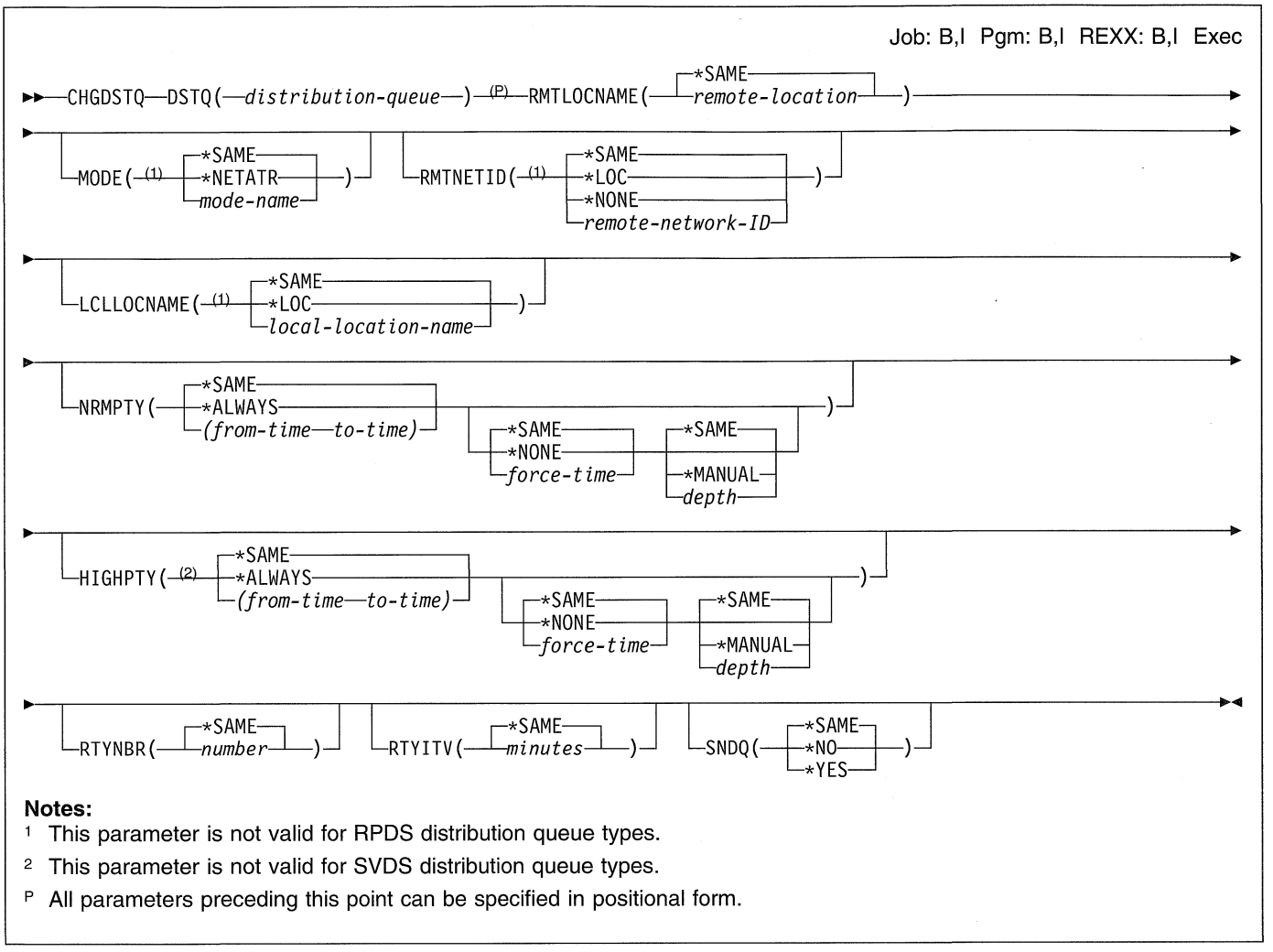
This command changes the DST password to the system-shipped default.

Example 2: Leaving the DST Password Unchanged

```
CHGDSTPWD  PASSWORD(*SAME)
```

This command does not change the DST password.

CHGDSTQ (Change Distribution Queue) Command



Notes:

1 This parameter is not valid for RPDS distribution queue types.

2 This parameter is not valid for SVDS distribution queue types.

P All parameters preceding this point can be specified in positional form.

Purpose

The Change Distribution Queue (CHGDSTQ) command changes an entry in the distribution services queue table. Distribution queues are used to store distributions before they are sent or forwarded to other systems.

The CHGDSTQ command does not provide interactive display support. This is provided by the Configure Distribution Services (CFGDSTSRV) command. More information about configuring a distribution network is in the *Distribution Services Network Guide*.

If connection information (remote location name, mode, remote network identifier, and local location name) is changed and a SNADS (SNA distribution services) sender job is active (but not sending), the SNADS sender job is ended and a new job is started to ensure that the job name always reflects the remote location name.

Distribution queue names are translated to the graphic character set and code page 930 500, using the job's coded character set identifier (CCSID).

Restrictions:

1. This command is shipped with public *EXCLUDE authority, and the QPGMR and QSYSOPR user profiles have private authorities to use the command.
2. The combination of remote location name, mode, remote network identifier, and local location name must be unique within the type of distribution queue. This combination does not need to be unique within the system, for SNA distribution services (SNADS) distribution queues in the distribution services queue table (SNADS-type distribution queues), and for SystemView distribution services (SVDS) distribution queues (SVDS-type distribution queues). The default value *LOC, which can be specified on the RMTNETID parameter and the LCLLOCNAME parameter, and the default value *NETATR, which can be specified on the MODE parameter, represent any possible values that the system determines are not already configured for another SNADS or SVDS distribution queue of the same type.
3. A unique remote location name must be specified for each RPDS-type distribution queue in the queue table.

RPDS queues do not use modes, remote network identifiers, or local location names.

4. Configuration in the routing table is not required for SVDS-type distribution queues. SVDS queues may be configured optionally in the SNADS routing table. However, normal SNADS mail can neither be routed to change management queues nor be received through change management connections, and change management connections can neither be routed to SNADS queues nor be received through SNADS connections.
5. SVDS-type distribution queues can support only a single queue view (the queue is not divided into normal and priority halves). For configurations and operations purposes, only the normal queue is specified.
6. Connection information cannot be changed when the queue has a status of "Sending" or when queue entries have the status of "Suspend" or "Pending." Wait for the status of the queue or queue entry to be no longer active or, as an alternative, use the Initialize Distribution Queue (INZDSTQ) command to reset this condition.
7. SVDS-type distribution queues cannot be changed when a receiver is active or when distributions have been received and the sender has not acknowledged receiving the confirmation. Wait for the remote system to complete sending or, as an alternative, use the Initialize Distribution Queue (INZDSTQ) command to reset this condition.
8. Messages that report errors about distribution queues may display or print different characters than the user entered for the distribution queue name because of internal system transformations. Similarly (depending on the language used for the work station), the internal value for a distribution queue name may differ from the characters shown on the Work with Distribution Queue (WRKDSTQ) command. An error may be reported if the character-string value specified for the DSTQ parameter does not match the rules for an internal distribution queue value or if it does not match the internal value for any defined distribution queue (ignoring case differences).

Required Parameters

DSTQ

Specifies the name of the distribution queue being changed.

RMTLOCNAME

Specifies the name of the remote location where distributions are sent from this distribution queue. The remote location name must be configured in the device description of the device used when sending distributions to another system from this distribution queue.

***SAME:** The name of the remote location does not change.

remote-location: Specify a maximum of 8 characters for the name of the remote location.

Optional Parameters

MODE

Specifies the name of the mode that defines the sessions on the device used by the distribution queue. This parameter is not applicable to RPDS-type distribution queues.

***SAME:** The name of the mode does not change.

***NETATR:** The mode name defined in the network attributes is used.

mode-name: Specify a maximum of 8 characters for the name of the mode. The mode name cannot be CPSVCMG or SNASVCMG; these mode names are reserved for system use.

RMTNETID

Specifies the remote network identifier of the remote network to which this distribution queue sends distributions. This parameter is not applicable to RPDS-type distribution queues.

***SAME:** The remote network identifier does not change.

***LOC:** The remote network identifier defined in the device description used by this distribution queue is used.

***NONE:** No remote network identifier is specified.

remote-network-ID: Specify the remote network identifier.

LCLLOCNAME

Specifies the name used to identify the local system to remote systems in the network. It is recommended that the name be the same as the local system name. This parameter is not applicable to RPDS-type distribution queues.

***SAME:** The local location name does not change.

***LOC:** The local location name defined in the device description used by this distribution queue is used.

local-location-name: Specify a maximum of 8 characters for the local location name.

NRMPTY

Specifies the queue sending conditions for distributions having a service level of data low.

Element 1: Send Time

Send time is the time period during which queued distributions of this priority are sent from this distribution queue. If you do not enter a time period, the transmissions are controlled by queue depth and are not related to time. The time must be specified in the 24-hour format `hhmm` where `h` = hours and `m` = minutes.

***SAME:** The send time does not change.

***ALWAYS:** Distributions of this priority are sent from this distribution queue regardless of the time of day.

from-time: Specify the time of day at which the system starts sending distributions of this priority from this distri-

bution queue if the value specified for send depth is reached. If from time is specified, to time must also be specified.

to-time: Specify the time of day at which the system stops sending distributions of this priority from this distribution queue. If to time is specified, from time must also be specified.

Element 2: Force Time

Force time is the time during which distributions of this priority are sent regardless of queue depth. If *ALWAYS is specified for the send time, the force time can be set to any time of day. If you specify a specific to-time and from-time for the send time, the force time must occur within that time period. The time must be specified in the 24-hour format hhmm where h = hours and m = minutes.

***SAME:** The force time does not change.

***NONE:** No force time is specified.

force-time: Specify the time at which distributions of this priority are sent regardless of queue depth.

Element 3: Depth

Queue depth specifies the number of distributions of this priority that are on the queue before sending can begin.

***SAME:** The queue depth value does not change.

***MANUAL:** Distributions of this priority are sent only when an operator manually sends them using the Work with Distribution Queue (WRKDSTQ) command or the Send Distribution Queue (SNDDSTQ) command.

depth: Specify the number of distributions of this priority that are on this distribution queue before any are sent. Valid values range from 1 through 999.

HIGHPTY

Specifies the queue sending conditions for distributions having a service level of fast, status, or data high. On this parameter, time must be specified in the 24-hour format hhmm where h = hours and m = minutes.

Element 1: Send Time

***SAME:** The send time does not change.

***ALWAYS:** Distributions of this priority are sent from this distribution queue regardless of the time of day.

from-time: Specify the time of day at which the system starts sending distributions of this priority from this distribution queue if the value specified for send depth is reached. If from time is specified, to time must also be specified.

to-time: Specify the time of day at which the system stops sending distributions of this priority from this distribution queue. If to time is specified, from time must also be specified.

Element 2: Force Time

***SAME:** The force time does not change.

***NONE:** No force time is specified.

force-time: Specify the time at which distributions of this priority are sent regardless of queue depth.

Element 3: Depth

***SAME:** The queue depth value does not change.

***MANUAL:** Distributions of this priority are sent only when an operator manually sends them using the Work with Distribution Queue (WRKDSTQ) command or the Send Distribution Queue (SNDDSTQ) command.

depth: Specify the number of distributions of this priority that are on this distribution queue before any are sent. Valid values range from 1 through 999.

RTYNBR

Specifies the maximum number of times the system attempts to resend distributions from this distribution queue after a failure occurs. This parameter applies to communications line failures and recoverable distribution failures on a remote system. The SNADS job serving this distribution queue ends when the number of retries is exceeded.

***SAME:** The maximum number of retry attempts does not change.

number: Specify the maximum number of times the system can attempt to resend distributions after a failure. Valid values range from 0 through 9999.

RTYITV

Specifies the interval (in minutes) between each retry attempt.

***SAME:** The retry interval does not change.

minutes: Specify the interval between retry attempts. Valid values range from 0 to 9999 minutes.

SNDQ

Specifies whether this distribution queue ignores the send time and depth values specified on the NRMPTY and HIGHPTY parameters and begins sending when a distribution is received from the SNADS system to which the queue sends its distributions.

This parameter is valid only if a SNADS-type distribution queue is being changed.

***SAME:** The send queue value does not change.

***NO:** Distributions are sent from this queue only when the queue's sending conditions are met.

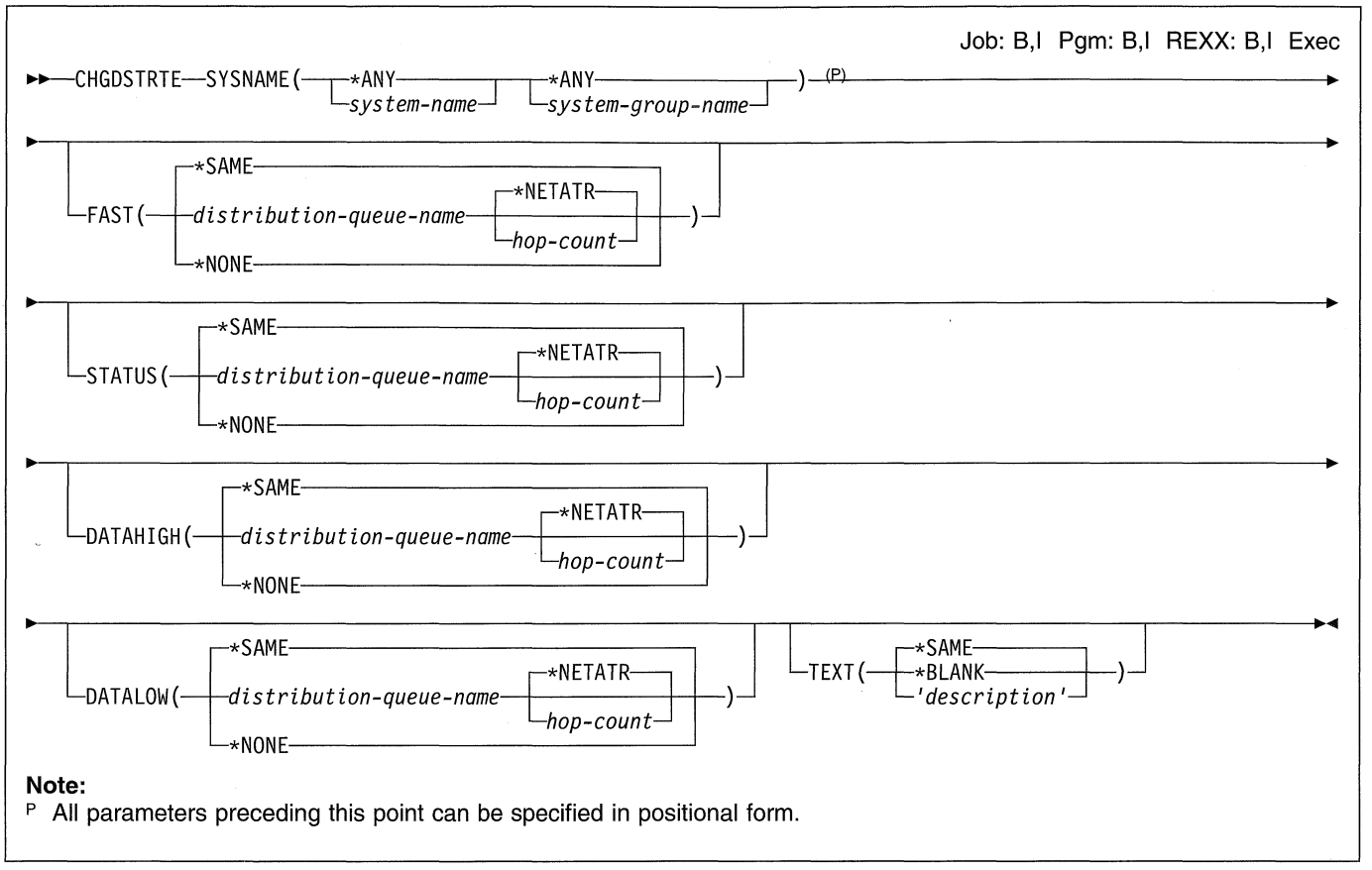
***YES:** This distribution queue begins sending when distributions are received from the SNADS system to which the queue sends its distributions regardless of the queue's other sending conditions. Distributions are automatically sent for manual queues (queues that have no specified depth variable).

Example

```
CHGDSTQ    DSTQ(CHICAGO)   RMTLOCNAME(NEWLU)   MODE(*NETATR)
```

This command changes the distribution queue named CHICAGO. The remote location name is changed to NEWLU and the mode is changed to the mode specified in the system network attributes.

CHGDSTRTE (Change Distribution Route) Command



Note:
P All parameters preceding this point can be specified in positional form.

Purpose

The Change Distribution Route (CHGDSTRTE) command changes an entry in the distribution services routing table. One or more service levels must be specified for each routing table entry.

Interactive display support is provided by the Configure Distribution Services (CFGDSTSRV) command. More information about configuring a distribution network is in the *Distribution Services Network Guide*.

System names, system group names, and distribution queue names are translated to the graphic character set and code page 930 500, using the job's coded character set identifier (CCSID).

Restrictions:

1. This command is shipped with public *EXCLUDE authority, and the QPGMR and QSYSOPR user profiles have private authorities to the command.
2. You must specify a value on at least one of the following parameters: FAST, STATUS, DATAHIGH, and DATALOW. You cannot change the value on all four parameters to *NONE.
3. An error occurs if a distribution route is changed to include a SystemView distribution services (SVDS) type of distribution queue with another type of distribution queue (such as SNA distribution services (SNADS) or VM/MVS bridge (RPDS)).

Required Parameter

SYSNAME
Specifies the system name and group name of the remote system for which a routing table entry is to be changed.

Element 1: System Name

***ANY:** *ANY is used for the system name. When *ANY and a system group name are specified, you change the routing table entry used to resolve a distribution destination that does not match a specific system name but matches a group name. Only one *ANY value is allowed for each group in the routing table.

system-name: Specify a maximum of 8 characters for the name of the remote system for which a routing table entry is to be changed.

Element 2: System Group Name

***ANY:** *ANY is used for the system group name. *ANY can be specified for the group name only if *ANY is also specified for the system name. When SYSNAME(*ANY *ANY) is specified, you change the routing table entry used to resolve a distribution destination that does not match any other routing table entries. Only one SYSNAME(*ANY *ANY) entry is allowed in the routing table.

system-group-name: Specify a maximum of 8 characters for the system group name. The system name and group name must be separated by at least one blank.

Optional Parameters

FAST

Specifies the distribution queue and maximum hop count to the destination system for fast service level distributions. The fast service level is the highest priority level.

The maximum hop count is the maximum number of times in a SNADS network that a distribution can be routed back and forth (hop) between the systems participating in the SNADS level routing, including the hop to the final destination system. The maximum hop count does not include the hops made by advanced peer-to-peer networking (APPN) routing. If the maximum number of hops is exceeded, the distribution is ended and an error is sent to the user who originally sent the distribution.

***SAME:** The distribution queue and maximum hop count to the destination system for fast service level distributions do not change.

***NONE:** No distribution queue is specified for distributions requiring a fast service level. Distributions requiring fast service cannot be routed using this routing table entry.

Element 1: Distribution Queue

distribution-queue-name: Specify the name of the distribution queue to which fast distributions using this routing entry are sent. The distribution queue must already exist and cannot be a DLS (document library services) type of queue.

Element 2: Maximum Hop Count

***NETATR:** The system network attribute value for the maximum hop count is used. The current system default value can be displayed using the Display Network Attributes (DSPNETA) command.

hop-count: Specify the maximum hop count. Valid values range from 1 through 255.

STATUS

Specifies the distribution queue and maximum hop count to the destination system for status service level distributions. The status service level is used for network status and other feedback information.

***SAME:** The distribution queue and maximum hop count to the destination system for status service level distributions do not change.

***NONE:** No distribution queue is specified for distributions requiring a status service level. Distributions requiring status service cannot be routed using this routing table entry.

Element 1: Distribution Queue

distribution-queue-name: Specify the name of the distribution queue to which status distributions using this routing entry are sent. The distribution queue must already exist and cannot be a DLS-type queue.

Element 2: Maximum Hop Count

***NETATR:** The system network attribute value for the maximum hop count is used. The current system value can be displayed using the Display Network Attributes (DSPNETA) command.

hop-count: Specify the maximum hop count. Valid values range from 1 through 255.

DATAHIGH

Specifies the distribution queue and maximum hop count to the destination system for data high service level distributions. The data high service level is used for high priority data traffic.

***SAME:** The distribution queue and maximum hop count to the destination system for data high service level distributions do not change.

***NONE:** No distribution queue is specified for distributions requiring a data high service level. Distributions requiring data high service cannot be routed using this routing table entry.

Element 1: Distribution Queue

distribution-queue-name: Specify the name of the distribution queue to which data high distributions using this routing entry are sent. The distribution queue must already exist and cannot be a DLS-type queue.

Element 2: Maximum Hop Count

***NETATR:** The system network attribute value for the maximum hop count is used. The current system value can be displayed using the Display Network Attributes (DSPNETA) command.

hop-count: Specify the maximum hop count. Valid values range from 1 through 255.

DATALOW

Specifies the distribution queue and maximum hop count to the destination system for data low service level distributions. The data low service level is used for most data traffic.

***SAME:** The distribution queue and maximum hop count to the destination system for data low service level distributions do not change.

***NONE:** No distribution queue is specified for distributions requiring a data low service level. Distributions requiring data low service cannot be routed using this routing table entry.

Element 1: Distribution Queue

distribution-queue-name: Specify the name of the distribution queue to which data low distributions using this routing entry are sent. The distribution queue must already exist and cannot be a DLS-type queue.

Element 2: Maximum Hop Count

***NETATR:** The system network attribute value for the maximum hop count is used. The current system value can be displayed using the Display Network Attributes (DSPNETA) command.

hop-count: Specify the maximum hop count. Valid values range from 1 through 255.

TEXT

Specifies text that briefly describes the distribution route. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

Note: Double-byte character set (DBCS) characters can be entered on this parameter.

***SAME:** The text does not change.

***BLANK:** The text is changed to blanks.

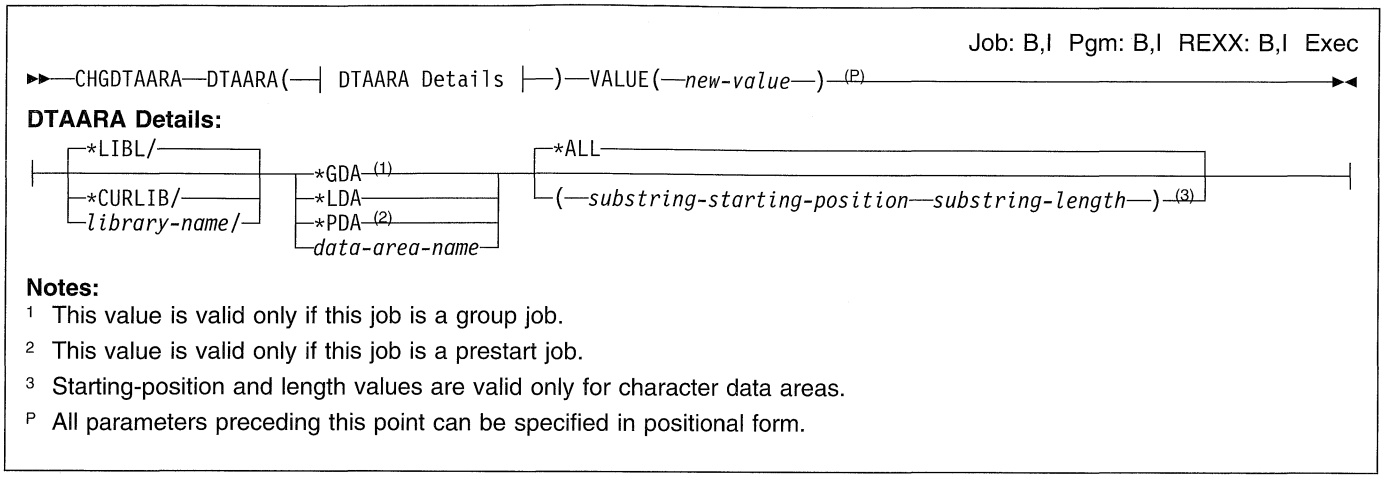
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGDSTRTE   SYSNAME(SYSTEMA)
  DATAHIGH(SYSTEMAFAST 2)
```

This command changes the distribution queue for distributions sent to SYSTEMA with a data high service level. The new distribution queue is SYSTEMAFAST with a hop count of 2.

CHGDTAARA (Change Data Area) Command



Notes:

1 This value is valid only if this job is a group job.

2 This value is valid only if this job is a prestart job.

3 Starting-position and length values are valid only for character data areas.

P All parameters preceding this point can be specified in positional form.

Purpose

The Change Data Area (CHGDTAARA) command changes the value of the local data area, the group data area, the program initialization parameter data area, or the specified data area stored in a library. This command does not change the data attributes or any of the object attributes of the data area. The new value must have the same type and a length less than or equal to the data area length or the specified substring length.

In group jobs, the data area specified may be the group data area (*GDA). This data area is automatically associated with the group and cannot be accessed from jobs outside the group. The length of this character data area is 512 bytes. More information about group jobs is in the *Work Management Guide*.

The local data area (*LDA) is a character data area 1024 bytes in length. It is automatically associated with your job and jobs outside the group cannot get access to it.

In prestart jobs, the data area specified may be the data area that contains program initialization parameter data (*PDA). This data area is automatically associated with your prestart job and other jobs cannot get access to it. The length of this character data area is 2000 bytes. More information about prestart jobs is in the *Work Management Guide*.

For character data areas, a substring of the data area may be changed without affecting the rest of the data area. This substring is defined by specifying the starting position and the length of the substring. In this case, the new value must have a length less than or equal to the substring length.

When the CHGDTAARA command is run to change a data area other than the local data area, group data area, or program initialization parameter data area, that data area is locked to the program during the change operation so that commands in other jobs cannot change or destroy it until the operation is completed. If the data area is shared with other jobs, and it is updated in steps involving more than one command in a job, the data area must be explicitly allocated to that job until all the steps have been performed. Data areas, other than the group data area, local data area, or program initialization parameter data area, can be explicitly allocated with the Allocate Object (ALCOBJ) command. No allocation is necessary when the CHGDTAARA command is run specifying *GDA, *LDA, or *PDA in the DTAARA parameter.

Restriction: To use this command, the user must have update authority for the data area being changed and read authority for the library in which it is stored. No specific authority is required for the local data area or the group data area.

Required Parameters

DTAARA

Specifies the qualified name of the data area whose value is being changed. It optionally specifies, for character data areas only, the starting position and length of the character string that is changed in the data area.

Element 1: Data Area Name

The name of the data area can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

***GDA:** The group data area associated with a group job is changed.

***LDA:** The local data area associated with a job is changed.

***PDA:** The program initialization parameter data area associated with a prestart job is changed.

data-area-name: Specify the name of the data area within which a character string is being changed.

Element 2: Starting Position of the Data Area

***ALL:** The entire data area is changed.

substring-starting-position: Specify the starting position of the data area that is changed.

Element 3: Length of the Data Area

substring-length: Specify the length of the data area that is changed.

VALUE

Specifies the new value stored in the data area. Specify a value that is valid for the data attributes specified in the data area's description. If TYPE(*CHAR) or TYPE(*LGL) is specified when the data area was created and the value specified here is numeric, the value must be enclosed in apostrophes. If TYPE(*DEC) was specified, the value must not be enclosed in apostrophes.

Examples

Example 1: Changing the Data Area Value

```
CHGDTAARA   DTAARA(MYLIB/MYDATA)  VALUE(GOODNIGHT)
```

This command changes the value of the data area named MYDATA in library MYLIB to GOODNIGHT. The data area must be for character data and must be 9 or more characters in length.

Example 2: Changing the Logical Value of the Data Area

```
CHGDTAARA   PAYROLLSW  '0'
```

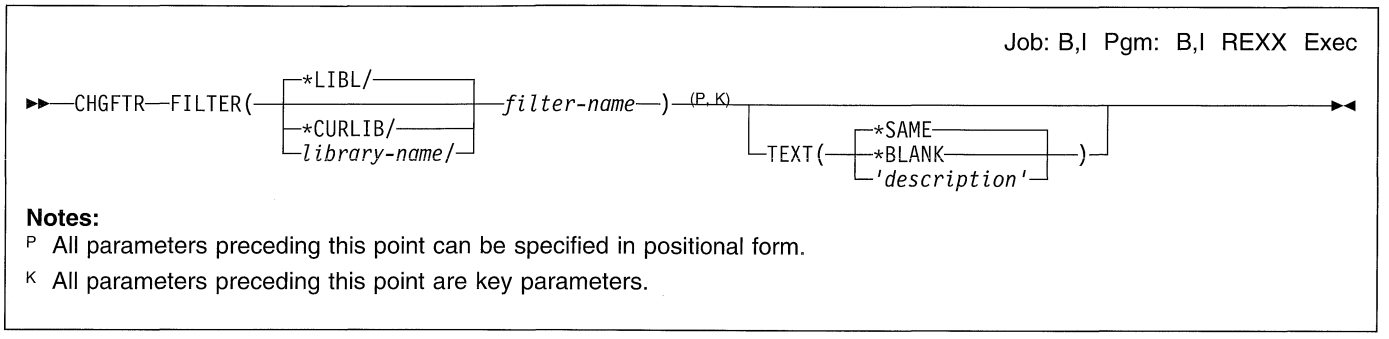
This command changes the logical value of the data area named PAYROLLSW to zero (0). The library search list is used to locate the data area.

Example 3: Changing Specific Characters of the Local Data Area

```
CHGDTAARA   DTAARA(*LDA  (5 4))  VALUE('TWO')
```

This command changes characters 5 through 8 of the user's local data area. Because the new value is shorter than the substring, it is padded with a blank.

CHGFTR (Change Filter) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Filter (CHGFTR) command allows you to change the text description of a filter object.

Required Parameter

FILTER

Specifies the qualified name of the filter whose description is being changed.

The name of the filter can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

filter-name: Specify the name of the filter that is to be changed.

Optional Parameter

TEXT

Specifies text that briefly describes the filter. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

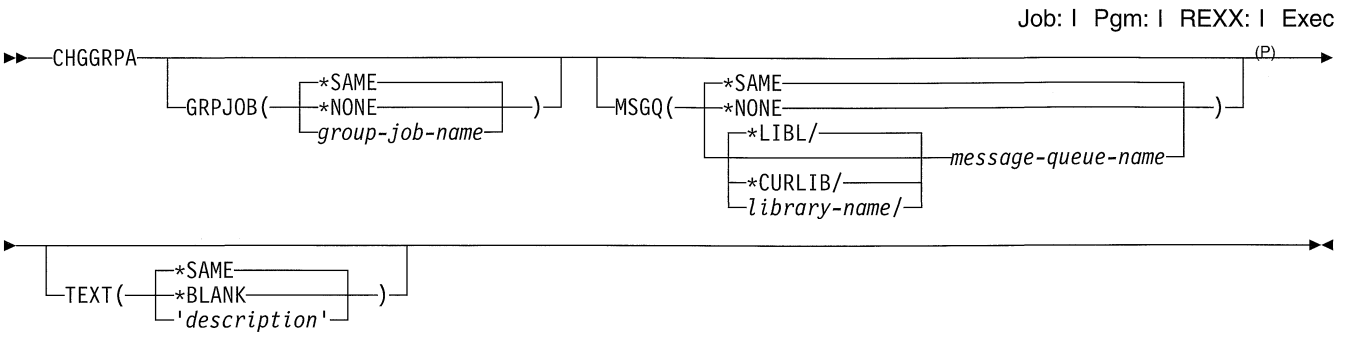
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGFTR  FILTER(MYLIB/MYFILTER)  TEXT('New text here')
```

This command changes the description of the filter called MYFILTER in the library MYLIB.

CHGGRPA (Change Group Attributes) Command



Note:

| P All parameters preceding this point can be specified in positional form.

Purpose

The Change Group Attributes (CHGGRPA) command changes the group attributes of a job. The following attributes can be changed:

- An interactive job can be changed to a group job.
- A group job can be changed to a nongroup, interactive job.
- A message queue can be associated with a group.
- A message queue can be disassociated from a group.
- A group job's descriptive text can be changed.

Optional Parameters

GRPJOB

Specifies the group job name assigned to the job in which this CHGGRPA command is used.

***SAME:** The value does not change.

***NONE:** The only active job in a group is changed to a nongroup interactive job, the group data area (*GDA) is deleted, and the group message queue (if there is one) is disassociated from the group; the mode of the disassociated message queue, however, does not change.

group-job-name: Specify the group job name that is assigned to the interactive job in which this command is used. This causes the job to become a group job and also causes the group data area (*GDA) to be created.

Specifying a group job name is valid only when the job is not a group job. Once a group job name is assigned to a job, that name cannot be changed (except by setting it to *NONE if it is the only active job in the group).

When a job becomes a group job by means of the CHGGRPA command, the group data area is created. This character type data area is 512 bytes long, and at first, it is filled with blanks. Jobs in a group may communicate with each other by using the group data area; it is accessible from any of the jobs in the group, but not from outside that group. The group data area is supported on the Change Data Area (CHGDTAARA), Retrieve Data Area (RTVDTAARA), and the Display Data Area (DSPDTAARA) commands by using the special value *GDA.

MSGQ

Specifies the qualified name of a message queue associated with a group.

***SAME:** The value does not change.

***NONE:** No message queue is associated with the group. If a message queue is currently associated with the group, this value disassociates the message queue from the group, but the mode does not change.

| The name of the message queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

message-queue-name: Specify the name of the message queue associated with the group. This message queue must exist, but not necessarily be allocated to the job when this CHGGRPA command is issued.

The message queue is available to the active job in the group. If one group job either starts or transfers to another group job, and the message queue is in break or notify mode in the job that is issuing the Transfer Group Job (TFRGRPJOB) command, the message queue is placed in the same mode as that of the group job to which it is transferred. If the message queue is not in break or notify mode in the job that is issuing the TFRGRPJOB command, then no special processing is

done for the message queue even though it has been associated with the group.

It is possible to have two groups associated with the same work station when a secondary interactive job exists (the Transfer Secondary Job (TFRSECJOB) command is entered, or option 1 on the system request menu is entered). If two groups are associated with a work station and both groups specify the same message queue, only one of the groups can have the message queue in break or notify mode at a time. For both groups to share the same group message queue, the message queue mode must be set to either the hold mode or to the default mode before the TFRSECJOB command or System Request option 1 is issued.

Note: Work station message queues, and the QHST, QSRV, and QCHG system message queues are not valid for this parameter.

TEXT

Specifies text that briefly describes the group job. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

Note: This text appears on the Group Job Selection display for the TFRGRPJOB command.

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Examples

Example 1: Changing an Interactive Job to a Group Job

```
CHGGRPA  GRPJOB(GROUPJ1)  MSGQ(QGPL/GRPMSGQ)
```

An interactive job is first changed to a group job before it is allowed to create other group jobs at the work station. This command changes a work station job into a group job with the group job name GROUPJ1; a message queue is associated with the group and (as long as it is in break or notify mode in the job issuing a TFRGRPJOB or End Group Job (ENDGRPJOB) command) is allocated to the active job in the group. The group data area (*GDA) is also created.

Example 2: Changing a Group Job to a Nongroup Job

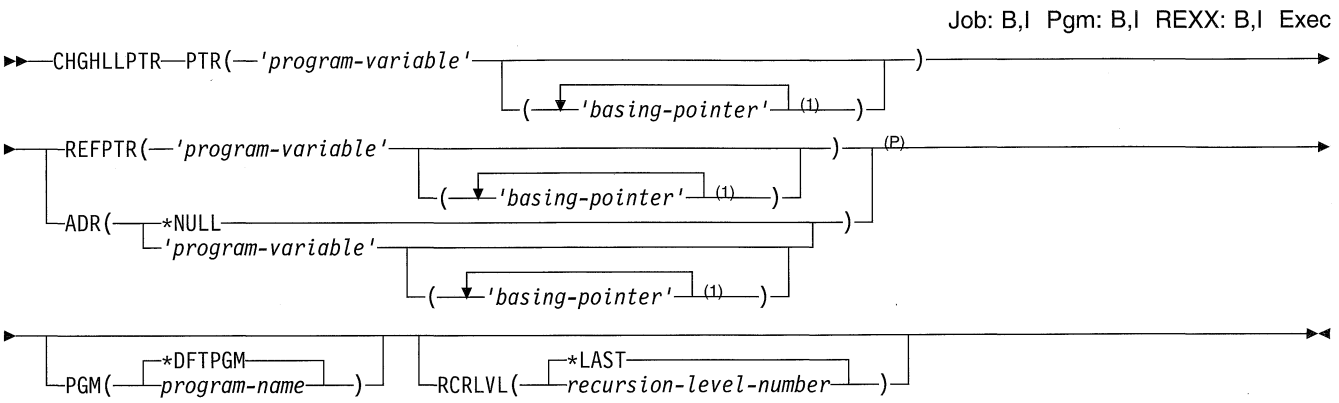
```
CHGGRPA  GRPJOB(*NONE)
```

Assume that the job in the previous example issues this command. Assume also that the job is the only active job in the group. Group job GROUPJ1 is changed into a nongroup, interactive job. The group job name is removed, and the group data area (*GDA) is deleted. The change from a group job into a nongroup job occurs only if it is the only active job in the group.

Additional Considerations

- It is not necessary to change the group job into a nongroup job; it could have remained a group job until sign-off.
- The job's work station message queue is automatically passed (if the message queue is in *BREAK or *NOTIFY mode) from job to job through System Request and TFRGRPJOB.

CHGHLLPTR (Change High-Level Language Pointer) Command



Notes:
1 A maximum of 5 repetitions
P All parameters preceding this point can be specified in positional form.

Purpose

The Change High Level Language Pointer (CHGHLLPTR) command allows the user to change or copy a pointer variable declared in a high-level language (HLL) program.

The following functions can be requested:

- Copy the value of one HLL pointer into another HLL pointer variable (REFPTR parameter).
- Set an HLL pointer variable to point to a variable in the program (ADR parameter).
- Set the value of the HLL pointer variable to a null pointer (ADR parameter).

HLL pointers are pointer variables that are declared in the source of an HLL program; they are contrasted with internal compiler-created pointers. Compiler-created pointers are pointer variables that are declared in the machine-interface (MI) program, by a compiler, to support various processing functions. Not all languages allow the user to declare HLL pointers.

Restrictions:

1. This command is valid only in debug mode. To start debug mode, refer to the STRDBG (Start Debug) command.
2. This command cannot be used if the user is servicing another job, and that job is on a job queue, or is being held, suspended, or ended.
3. This command cannot be used to change variables in a bound program.
4. This command cannot be used to change variables that are write-protected or within the system domain, unless the user has *SERVICE special authority.

Required Parameters

PTR

Specifies the name of the pointer (program variable) whose value is being changed, allowing the pointer to point to a different address.

Element 1: Program Variables

'program-variable': Specify the name of the pointer variable whose value is being changed. The pointer variable must be an HLL pointer (HLLPTR). A machine-interface object-definition-table-vector (MI ODV) number cannot be specified for the variable name, even though the variable that it defines is an HLL pointer. The name must be enclosed in apostrophes if it contains special characters.

If the pointer (program variable) is in an array, the subscripts representing the element in the array to be changed must be specified. Up to 132 characters may be specified for this pointer (program variable) entry. This includes any qualifiers, subscripts, embedded blanks, parentheses, and commas. It does not include the enclosing apostrophes when special characters are used. An integer, MI ODV number, or numeric variable name can be specified for a subscript. For more information on changing program variables during testing, refer to the *CL Programmer's Guide* and Appendix C, "Parameter Values Used for Testing and Debugging."

Some examples are:

```
PTR('PTR_ARRAY(2)')
PTR(P1)
```

Element 2: Basing Pointers

'basing-pointer': Specify up to five basing pointers. In some languages, the pointer (program-variable) may be based on a pointer variable. This set of values allows the user to specify the basing pointers for the pointer to

be changed. The basing pointers must be HLL pointers (HLLPTR). An MI ODV number cannot be specified for the name, even though the pointer that defines it is an HLL pointer. Each basing pointer name must be enclosed in apostrophes if it contains special characters. If the basing pointer is an array, the subscripts representing an element in the array must be specified. Up to 132 characters can be specified for a basing pointer name. This includes any qualifiers, subscripts, embedded blanks, parentheses, and commas. It does not include the enclosing apostrophes when special characters are used. For more information on the basing pointer value, refer to Appendix C, "Parameter Values Used for Testing and Debugging."

Some examples are:

```
PTR('P@(B,5)'  'Q@(I)')
PTR(PTR1 (BASEPTRA BASEPTRB))
```

REFPTR

Specifies the name of the pointer (program variable) whose value is being copied into the pointer identified in the PTR parameter.

Element 1: Program Variables

'program-variable': Specify the name of the pointer variable whose value is copied into the specified pointer. The pointer variable must be an HLL pointer (HLLPTR). An MI ODV number cannot be specified for the variable name, even though the variable that it defined is an HLL pointer. The name must be enclosed in apostrophes if it contains special characters. If the pointer (program variable) is in an array, the subscripts representing the element in the array to be copied must be specified. Up to 132 characters may be specified for this pointer (program variable) entry. This includes any qualifiers, subscripts, embedded blanks, parentheses, and commas. It does not include the enclosing apostrophes when special characters are used. An integer, MI ODV number, or numeric variable name can be specified for a subscript.

Some examples are:

```
REFPTR('PRT1(2,3)')
REFPTR(C2PTR)
```

Element 2: Basing Pointers

'basing-pointer': Specify up to five basing pointers. The pointer (program variable) may be based on a pointer variable. This set of values allows the user to specify the basing pointers for the pointer to be copied. The basing pointer must be an HLL pointer (HLLPTR). An MI ODV number cannot be specified for the name, even though the pointer that defines it is an HLL pointer. Each basing pointer name must be enclosed in apostrophes if it contains special characters. If the basing pointer is an array, the subscripts representing an element in the array must be specified. Up to 132 characters can be specified for a basing pointer name. This includes any qualifiers, subscripts, embedded blanks, parentheses, and commas. It does not include the enclosing apostrophes when special characters are used. For more information on the basing pointer value, refer to Appendix C, "Parameter Values Used for Testing and Debugging."

Some examples are:

```
REFPTR('P@(B,5)'  'Q@(I)')
REFPTR(PRT1 (BASEPTRA 'Q@(/1F)'))
```

ADR

Specifies the name of the program variable, if any, to which the specified HLL pointer is to point.

***NULL:** The HLL pointer is set to null; it no longer points to the address of any space object, and it no longer contains a space pointer or any type of pointer value.

Element 1: Program Variables

'program-variable': Specify the name of the variable to which the HLL pointer is to point. An HLL variable name must be entered; an MI ODV number cannot be used. The name must be enclosed in apostrophes if it contains special characters. If an array is specified without subscripts, the pointer is set to the address of the first element in the array. Up to 132 characters may be specified for this program variable entry. This includes any qualifiers, subscripts, embedded blanks, parentheses, and commas. It does not include the enclosing apostrophes when special characters are used. An integer, MI ODV number, or numeric variable name can be specified for a subscript. For more information on the program-variable value, refer to Appendix C, "Parameter Values Used for Testing and Debugging."

Some examples are:

```
ADR('VAR1(2,3)')
ADR(ARRAY2)
```

Element 2: Basing Pointers

'basing-pointer': Specify up to five basing pointers. In some languages, the program-variable may be based on a pointer variable. This set of values allows the user to specify the basing pointers for the variables to be addressed. The basing pointers must be HLL pointers (HLLPTR). An MI ODV number cannot be specified for the name, even though the pointer that defines it is an HLL pointer. Each basing pointer name must be enclosed in apostrophes if it contains special characters. If the basing pointer is an array, the subscripts representing an element in the array must be specified. Up to 132 characters can be specified for a basing pointer name. This includes any qualifiers, subscripts, embedded blanks, parentheses, and commas. It does not include the enclosing apostrophes when special characters are used. For more information on the basing pointer value, refer to Appendix C, "Parameter Values Used for Testing and Debugging."

Some examples are:

```
ADR('VAR(2,5)' 'PGMPTR(5)')
ADR(VAR5 (BASEPTRA BASEPTRB))
```

Optional Parameters

PGM

Specifies the name of the program that contains the pointer whose value is being changed.

***DFTPGM:** The program currently specified as the default program contains the pointer whose value is changed.

program-name: Specify the name of the program that contains the pointer whose value is being changed. The name must already have been specified in the STRDBG or Add Program (ADDPGM) command.

RCRLVL

Specifies which recursion level of the program contains the pointer whose value is being changed. Changes made to static variables automatically affect all recursion levels. Recursion level 1 is the first (or earliest) call of the program, recursion level 2 is the second recursion of the program, and so on to the last (most recent) recursion level in the stack. For example, if program A calls program B, and then program B calls program A, a new recursion level of program A is formed. If the first recursion level of program A contains the pointer being changed, RCRLVL(1) must be specified. Some high-level languages also allow recursive procedures. More information on these programs is in the appropriate high-level language manual.

***LAST:** The last, most recent, call of the specified program contains the pointer being changed.

recursion-level-number: Specify the number of the recursion level of the program containing the pointer being changed.

Examples

Example 1: Setting a Pointer to a Variable Address

```
CHGHLLPTR  PTR(STACKPTR)  ADR(FIRSTELEM)
```

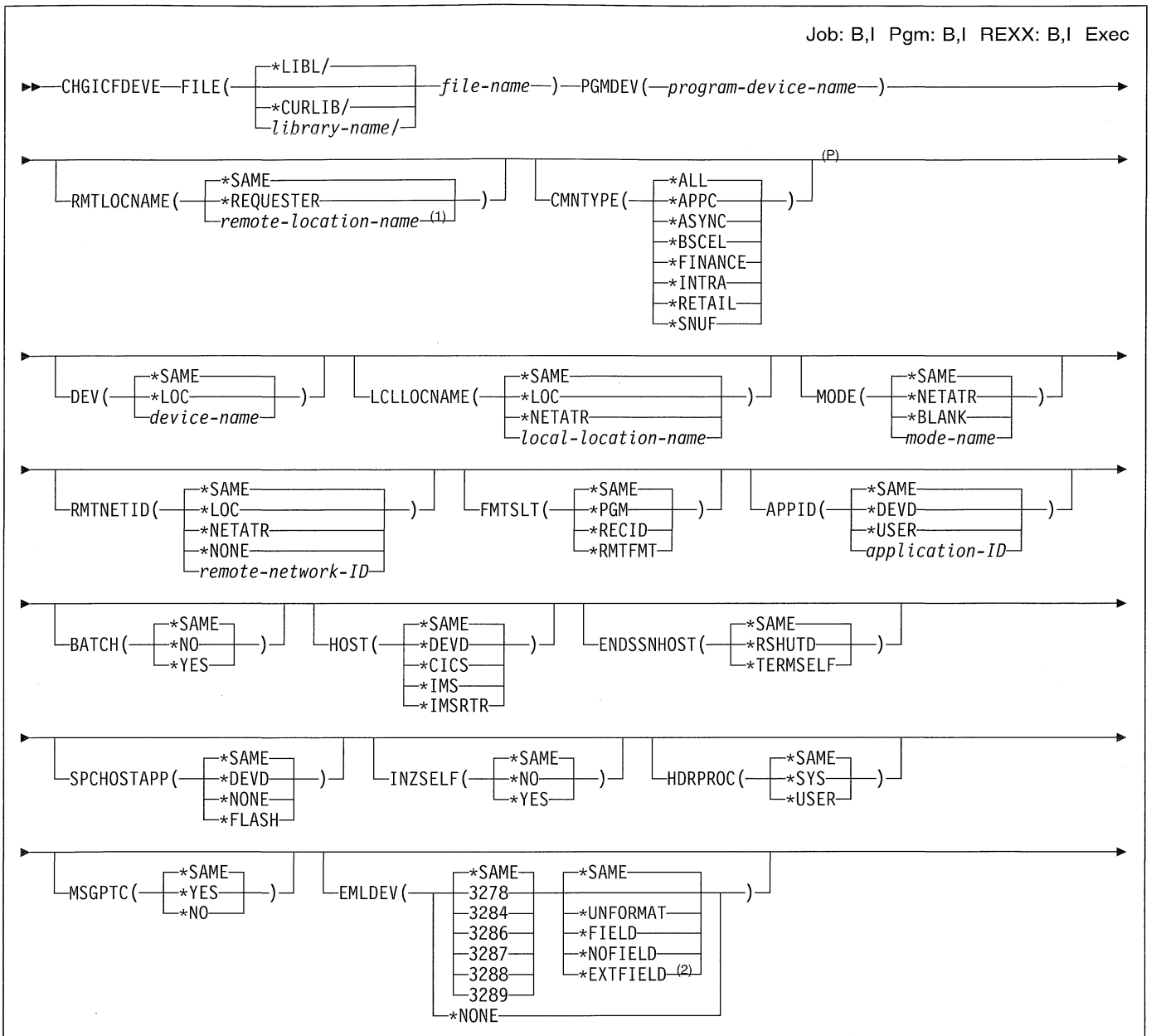
This command sets the pointer STACKPTR to the address of the variable FIRSTELEM.

Example 2: Copying the Value of a Pointer

```
CHGHLLPTR  PTR(NEXT@)  REFPTR(CURRENT@)
```

This command copies the value of the pointer CURRENT@ into the pointer NEXT@. The value copied could be a null pointer.

CHGICFDEVE (Change ICF Program Device Entry) Command

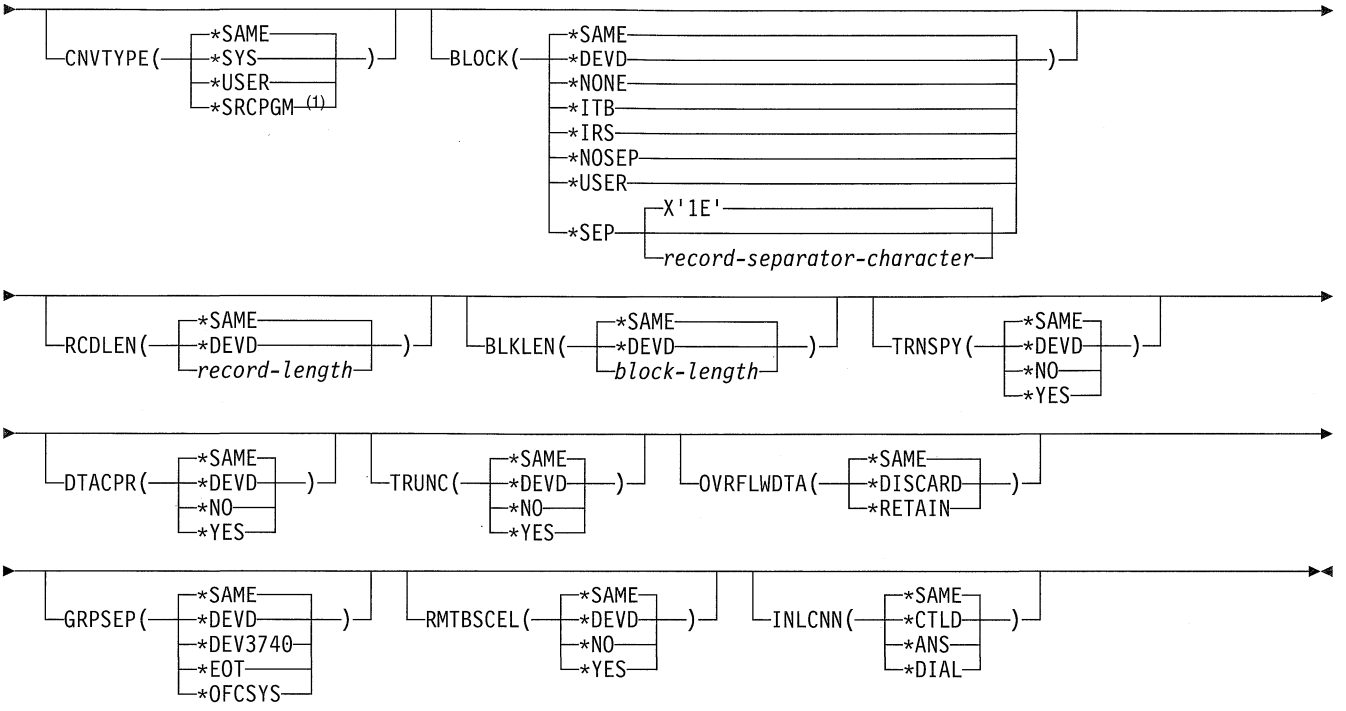


Notes:

1 This value cannot be specified with CNVTYPE (*SRCPGM).

P All parameters preceding this point can be specified in positional form.

2 The value *EXTFIELD is valid only when 3278 is specified.



Note:
1 This value can only be specified with RMTLOCNAME(*SAME) or RMTLOCNAME (*REQUESTER).

Purpose

The Change Intersystem Communications Function Program Device Entry (CHGICFDEVE) command changes the attributes of the specified program device entry in the specified Intersystem Communications Function (ICF) file.

Required Parameters

FILE

Specifies the qualified name of the ICF file for which the ICF program device entry is being changed.

The name of the ICF file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

file-name: Specify the name of the file that contains the program device entry to be changed.

PGMDEV

Specifies the program device entry name in the ICF file whose attributes are being changed. The program device entry must exist in the specified ICF file.

Optional Parameters

RMTLOCNAME

Specifies the remote location name of the system with which this object communicates.

More information on how the system uses the RMTLOCNAME parameter to select an APPC device description is in the *APPC Programmer's Guide*.

***SAME:** The value does not change.

***REQUESTER:** The name used to refer to the communications device through which the program is started is used. The session that is assigned when the program device is acquired is the same session in which the program start request is received. If the program is not started as a result of a program start request, the acquire operation of the program device fails. The target program uses *REQUESTER as the remote location name in the intersystem communications function (ICF) file to connect to the session that the source program used to send the program start request.

The *REQUESTER value can be specified on only one program device entry and is valid only for a target communication job. If *REQUESTER is specified in any other type of job, a message is sent.

remote-location-name: Specify the full name of a remote location. The remote location need not exist at the time this command is run, but must exist (be configured on the system as a device description or in the advanced

peer-to-peer networking (APPN) system for this remote location) when the program acquires the program device. This value cannot be specified with CNVTYPE(*SRCPGM).

CMNTYPE

Specifies which types of communications parameters show on the prompt screen. This parameter is used only for prompting purposes. The value specified for this parameter determines the subset of other parameters that are shown (prompted) for the user.

***ALL:** The parameters for all of the communications types appear in the prompt.

***APPC:** The advanced program-to-program communications (APPC) parameters appear in the prompt.

***ASYNC:** The asynchronous (ASYNC) parameters appear in the prompt.

***BSCEL:** The binary synchronous communications equivalence link (BSCEL) parameters appear in the prompt.

***FINANCE:** The finance parameters appear in the prompt.

***INTRA:** The intrasystem (INTRA) parameters appear in the prompt.

***RETAIL:** The retail parameters appear in the prompt.

***SNUF:** The SNA upline facility (SNUF) parameters appear in the prompt.

DEV

Specifies the communications device used in the remote location. This parameter can only be specified for APPC, FINANCE, RETAIL, INTRA, and SNUF communications types.

More information on how the system uses the DEV parameter to select an APPC device description is in the *APPC Programmer's Guide*.

***SAME:** The value does not change.

***LOC:** The device associated with the remote location is used. If several devices are associated with the remote location, the system determines which device is used.

device-name: Specify the name of a communications device associated with the remote location. If the device name is not valid for the remote location, a message is sent when the program device entry is acquired. More information on device names is in the *APPC Programmer's Guide*.

LCLLOCNAME

Specifies the local location name.

This parameter applies to the APPC communications type only and is ignored for all other communications types.

More information on how the system uses the LCLLOCNAME parameter to select an APPC device description is in the *APPC Programmer's Guide*.

***SAME:** The value does not change.

***LOC:** The device associated with the remote location is used. If several devices are associated with the remote location, the system determines which device is used.

***NETATR:** The LCLLOCNAME value specified in the system network attributes is used.

local-location-name: Specify the local location name associated with the program device entry. The local location name is specified only in APPC to indicate a specific local location name for the remote location. If the local location name is not valid for the remote location, or for the remote location and device, an escape message is sent when the program device entry is acquired.

MODE

Specifies the mode name used. This parameter applies only to the APPC communications type and is ignored for all other communication types.

***SAME:** The value does not change.

***NETATR:** The mode name specified in the network attributes is used.

***BLANK:** The mode name consisting of 8 blank characters is used.

mode-name: Specify a mode name for the APPC communications device. If the mode name is not valid for any combination of remote location device, local location, and remote network ID, an escape message is sent when the program device entry is acquired.

RMTNETID

Specifies the remote network identifier (ID) used with the remote location. This parameter applies to the APPC communications type only and is ignored for all other communications types.

More information on how the system uses the RMTNETID parameter to select an APPC device description is in the *APPC Programmer's Guide*.

***SAME:** The value does not change.

***LOC:** The remote network identifier (ID) associated with the remote location is used. If several remote network IDs are associated with the remote location, the system determines which remote network ID is used.

***NETATR:** The RMTNETID value specified in the system network attributes is used.

***NONE:** No remote network identifier (ID) is used.

remote-network-ID: Specify a remote network ID for the APPC communications device.

FMTSLT

Specifies the record format selection used for input operations.

***SAME:** The value does not change.

***PGM:** The program determines which record formats are selected. If an input (read) operation with a record format name is specified, that format is selected. If an input operation without a record format is specified, the default format (the first record format in the file) is selected. This also means that if there are any record identification (RECID) parameters specified in the data description specifications (DDS) for the file, or if any remote formats are received, they are not taken into consideration when the record is selected.

***RECID:** The record identification (RECID) keywords specified in the DDS for the file are used for record selection. If there are no RECID keywords in the file, an error message is sent, the acquire operation of the program device ends, and the device is not acquired.

***RMTFMT:** The remote format names received from the sending system are used for record selection. If the device is not an APPC or INTRA device and *RMTFMT is specified, a run time error occurs when the program device entry is acquired.

APPID

Specifies (in characters) the Virtual Telecommunications Access Method (VTAM) identifier (ID) of the Customer Information Control System for Virtual Storage (CICS/VS) or Information Management System/Virtual Storage (IMS/VS) host subsystem sent with the sign-on message. This parameter applies to the SNUF communications type only and is ignored for all other communications type.

***SAME:** The value does not change.

***DEVD:** The application identifier specified in the device description is sent without the sign-on message.

***USER:** The application program can send messages or a logon to the host. This is valid only when using the 3270 program interface.

application-ID: Specify the application identifier sent with the sign-on message.

BATCH

Specifies, for both CICS/VS and IMS/VS, whether this session is used for batch jobs. This parameter applies to the SNUF, RETAIL and INTRA communications types only and is ignored for all other communications types.

***SAME:** The value does not change.

***NO:** Batch jobs do not occur.

***YES:** Batch jobs occur.

HOST

Specifies the host or remote subsystem with which this session communicates. This parameter applies to the SNUF communications type only and is ignored for all other communications types.

***SAME:** The value does not change.

***DEVD:** The host system specified in the device description is used.

***CICS:** The session communicates with CICS/VS.

***IMS:** The session communicates with IMS/VS.

***IMSRTR:** The session communicates with IMS/VS using the ready-to-receive option.

ENDSSNHOST

Specifies how SNUF ends the session with the host. This parameter is valid only for SNUF communications.

***SAME:** The value does not change.

***RSHUTD:** SNUF sends a request-shut-down command to the host.

***TERMSELF:** SNUF sends a terminate-self command to the host. It may be necessary to specify this value if the value *RSHUTD fails to end a session with a non-IBM host.

SPCHOSTAPP

Specifies whether SNUF customizes support for special host applications outside the CICS or IMS application layer.

***SAME:** The value does not change.

***DEVD:** The special host application specified in the device description is used.

***NONE:** SNUF does not customize support for special host applications.

***FLASH:** SNUF customizes support for the Federal Link Access for Secondary Half-sessions (*FLASH) protocol application.

INZSELF

Specifies whether a formatted INIT-SELF is built in place of the unformatted sign-on normally sent by SNUF to the host.

***SAME:** The value does not change.

***NO:** The unformatted default sign-on provided by SNUF is used.

***YES:** The formatted INIT-SELF provided by SNUF is used.

HDRPROC

Specifies, for both CICS/VS and IMS/VS, whether received function management headers are passed to the application program. This parameter applies to the SNUF communications type only and is ignored for all other communications types.

***SAME:** The value does not change.

***SYS:** SNA upline facility (SNUF) removes function management headers before passing data to the program.

***USER:** Function management headers are passed with the data to the program.

MSGPTC

Specifies, for both CICS/VS and IMS/VS, whether message protection is used for this session. This parameter applies to the SNUF communications type only and is ignored for all other communications types.

***SAME:** The value does not change.

***YES:** Message protection is used. SNUF saves messages until they are responded to and tries synchronization again if errors occur. *YES is valid only when BATCH(*NO) is also specified.

***NO:** Message protection is not used.

EMLDEV

Specifies that the program device entry is used to send and receive data streams to and from specific types of 3270 display or printer devices being emulated. This parameter consists of an emulation device type and an emulation device data format. The emulation device data format specifies the format of the type 3270 data stream being sent or received. A 20- or 32-byte common header that contains type 3270 command and data flow information is located at the start of the I/O buffer that is sending or receiving the type 3270 data stream. This parameter applies to the SNUF communications type only. This parameter can be specified as a list of two values (elements) or as a single value (*NONE).

Element 1: Type of Device

***SAME:** The value does not change.

3278: The data stream is for a 3279, 3278, or 3277 display device.

3284: The data stream is for a 3284 printer.

3286: The data stream is for a 3286 printer.

3287: The data stream is for a 3287 printer.

3288: The data stream is for a 3288 printer.

3289: The data stream is for a 3289 printer.

Element 2: Format of the Data Stream

***SAME:** The value does not change.

***UNFORMAT:** An unformatted 3270 data stream is sent or received. The application program must translate the data stream into a display or printer image.

***FIELD:** A formatted 3270 data stream is sent or received. The formatted 3270 data stream contains a display or printer image followed by field definitions. The field definitions indicate the location and characteristics of each field.

***NOFIELD:** A formatted 3270 data stream that has no field definitions but contains a display or printer image is sent or received.

***EXTFIELD:** A formatted 3270 data stream contains extended field attribute information. The extended field attribute information is in the field definitions which follow the display image. The field definitions indicate the location and characteristics of each field. The value *EXTFIELD is valid only when 3278 is specified for the type of device on the EMLDEV parameter.

Note: If *FIELD, *NOFIELD, or *EXTFIELD is specified, BATCH(*NO) must also be specified.

Other Single Values

***NONE:** This program device entry is not used for sending and receiving 3270 data streams.

CNVTYPE

Specifies the conversation type for which the application program is designed. This parameter only applies to the APPC communications type, and is ignored for all other communications types.

More information on the APPC communications type is in the *APPC Programmer's Guide*.

***SAME:** The value does not change.

***SYS:** The advanced program-to-program communications (APPC) mapped conversation support is used.

***USER:** The advanced program-to-program communications (APPC) basic conversation support is used.

***SRCPGM:** The target program accepts the conversation type specified by the source program. If this value is specified, RMTLOCNAME(*REQUESTER) must also be specified.

BLOCK

Specifies whether the system or the user controls whether records are combined into blocks when they are sent. This parameter applies to the BSCEL communications type only and is ignored for all other communications types.

With this parameter, one of the following conditions of record formatting can be specified:

- No blocking or deblocking: The record format described in the DDS is the format used for both the record and the block.
- User blocking and/or deblocking: Give the BSC controls needed to describe the record format of the system.
- System blocking with record separator characters: Specify the record separator character used by the system to determine record boundaries in the block.
- System blocking of fixed-length records: The system uses fixed-length records, and blocks and/or deblocks them accordingly.

If a parameter value other than *NONE or *USER is specified, records are blocked as required by the system for output and are deblocked on input.

Element 1: Blocking Option

***SAME:** The value does not change.

***DEVD:** The block option specified in the device description is used.

***NONE:** Blocking or deblocking is not done by the system.

***ITB:** The records are blocked or deblocked based on the location of an intermediate text block (ITB) control character. For input files, a record is delimited by locating the next intermediate text block character. An end-of-text or end-of-transmission block character is used as an intermediate text block character to delimit a block. For output files, an ITB character is added after the record. If it is the last character of the block, the ITB is replaced by an end-of-text or end-of-transmission block character.

***IRS:** The records are blocked or deblocked based on the location of an interrecord separator (IRS) character. For input files, a record is delimited by locating the next IRS character. For output files, an IRS character is added after the record.

***NOSEP:** No record separator character is contained in the transmission block sent to or received from the device. The system blocks and deblocks the records by using a fixed-length record, as specified in the DDS format specifications.

***USER:** The program provides all the control characters (including record separator characters, binary synchronous communications (BSC) framing characters, and transparency characters) necessary to send records. More information about the device and binary synchronous communications equivalence link (BSCEL) support characteristics is in the *BSC Equivalence Link Programmer's Guide*.

***SEP:** The records are blocked or deblocked, based on the location of a record separator character that is specified by the user. For input files, a record is delimited by locating the next record separator character. For output files, a record separator character is added following the record.

Element 2: Record Separator

X '1E': The record separator character X '1E' is used.

record-separator-character: Specify a unique 1-byte record separator character. The record separator character can be specified as 2 hexadecimal characters, as in BLOCK(*SEP X'FD'), or as a single character by specifying a value ranging from 0 through 9 or A through Z, as in BLOCK(*SEP A).

The following are BSC control characters that cannot be used as record separator characters:

EBCDIC	ASCII	BSCEL Control
X'01'	X'01'	SOH (start-of-header)
X'02'	X'02'	STX (start-of-text)
X'03'	X'03'	ETX (end-of-text)
X'10'	X'10'	DLE (data-link escape)
X'1D'	X'1D'	IGS (interchange group separator)
X'1F'	X'1F'	ITB (intermediate text block)
X'26'	X'17'	ETB (end-of-transmission block)
X'2D'	X'05'	ENQ (enquiry)
X'32'	X'16'	SYN (synchronization)
X'37'	X'04'	EOT (end-of-transmission)
X'3D'	X'15'	NAK (negative acknowledgment)

RCDLEN

Specifies the maximum record length (in bytes) for data sent and received. This parameter applies to the BSCEL and the SNUF communications types only and is ignored for all other communications types.

***SAME:** The value does not change.

***DEVD:** The record length specified in the device description is used. If a record is longer than the specified record length, a run time error occurs when the record is sent or received.

record-length: Specify the maximum record length (in bytes) to use with this device file. The value must be at least the size of the largest record sent. If a record is longer than the specified record length, a run time error occurs when the record is sent or received. Valid values range from 1 through 32767 bytes for SNUF communications. For BSCEL communications, the maximum record length is 8192 bytes.

BLKLEN

Specifies the maximum block length (in bytes) for data sent. This parameter applies to the BSCEL and the SNUF communications types and is ignored for all other communications types.

***SAME:** The value does not change.

***DEVD:** The block length specified in the device description is used.

block-length: Specify the maximum block length (in bytes) of records sent. The value must be at least the size of the largest record sent. Valid values range from 1 through 32767 for SNA upline facility (SNUF). For binary synchronous communications equivalence link (BSCEL) communications, the maximum block length is 8192.

TRNSPY

Specifies whether data is sent in transparent text mode. Transparent text mode allows all 256 extended binary-coded decimal interchange code (EBCDIC) character codes to be sent. Use this function when sending packed or binary data fields. This parameter applies to the BSCEL communications type only and is ignored for all other communications types.

***SAME:** The value does not change.

***DEVD:** The text transparency option specified in the device description is used.

***NO:** Text transparency is not used.

***YES:** Text transparency is used, which permits the transmission of all 256 EBCDIC character codes. *YES is valid only when BLOCK(*NONE), BLOCK(*NOSEP), or BLOCK(*USER) is specified.

Note: Transparency of received data is determined by the data stream; therefore, this parameter is not relevant for received data. If TRNSPY(*YES) is specified with BLOCK(*USER), the BSCEL communications type ignores the transparency indicator during write operations. Correct controls must be given with the data to get transparent transmission of data. For example, the data-link escape (DLE) and start-of-text (STX) character control characters must first be specified; the system gives the remaining control characters for transparent sending of data.

DTACPR

Specifies whether data compression is performed.

Note: DTACPR(*YES) cannot be specified if TRNSPY(*YES) is specified. This parameter applies to the BSCEL communications type only and is ignored for all other communications types.

***SAME:** The value does not change.

***DEVD:** The data compression option specified in the device description is used.

***NO:** No data compression or decompression occurs.

***YES:** Data is compressed for output and decompressed for input.

TRUNC

Specifies whether trailing blanks are removed from output records. This parameter is for the BSCEL communications type only and is ignored for all other communications types.

***SAME:** The value does not change.

***DEVD:** The truncation option specified in the device description is used.

***NO:** Trailing blanks are not removed from output records.

***YES:** Trailing blanks are removed from output records.

Note: TRUNC(*YES) cannot be specified if BLOCK(*NOSEP) or BLOCK(*ITB) is specified. If TRUNC(*YES) is specified when DTACPR(*YES) or BLOCK(*USER) is specified, then truncation is ignored.

OVRFLWDTA

Specifies whether overflow data is discarded or retained.

***SAME:** The value does not change.

***DISCARD:** Overflow data is not kept.

***RETAIN:** Overflow data is kept.

GRPSEP

Specifies a separator for groups of data (for example, data sets and documents). This parameter applies to the BSCEL communications type only and is ignored for all other communications types.

***SAME:** The value does not change.

***DEVD:** The group separator option specified in the device description is used.

***DEV3740:** A null record (STXETX) is used as a data group separator.

***EOT:** An end-of-transmission (EOT) BSC control character is used as a data group separator.

***OFCSYS:** A block sent that ends with the BSC control character ETX (end-of-text) is used as a data group separator.

RMTBSCEL

Specifies the type of BSCEL session with the remote system. This parameter applies to the BSCEL communications type only and is ignored for all other communications types.

***SAME:** The value does not change.

***DEVD:** The RMTBSCEL option specified in the device description is used.

***NO:** The remote system cannot recognize BSCEL commands or messages. In most cases, *NO is used when communicating with remote systems such as a 3741 Data Entry Station, an Office System 6, a 5230 Data Collection System, or a System/38.

***YES:** The remote system can recognize the BSCEL transaction starting commands, transaction ending commands, and online messages. In most cases, *YES indicates that the remote system is another AS/400 system, System/38, System/36, or a System/34 with BSCEL support.

INLCNN

Specifies the method used to make a connection on the line for the session being acquired. This parameter applies to the binary synchronous communications equivalence link (BSCEL) communications types only.

***SAME:** The value does not change.

***CTLD:** The initial connection value specified in the controller description is used.

***ANS:** The remote system starts the call and the local system answers the call.

***DIAL:** The local system starts the call.

Examples

Example 1: Changing the Record Format Selection

```
CHGICFDEVE  FILE(ICFTEST)  PGMDEV(BSCEL2)
  RMTLOCNAME(BSCNYC)  FMTSLT(*RECID)
```

This command changes the program device entry named BSCEL2 in ICF file ICFTEST. The program device is changed to attributes of FMTSLT(*RECID). The remote location name is changed to BSCNYC.

Example 2: Changing the Conversation Type

```
CHGICFDEVE  FILE(QGPL/ICF1)  PGMDEV(APPC1)
  RMTLOCNAME(*REQUESTER)
  FMTSLT(*RMTFMT)  CNVTYPE(*SYS)
```

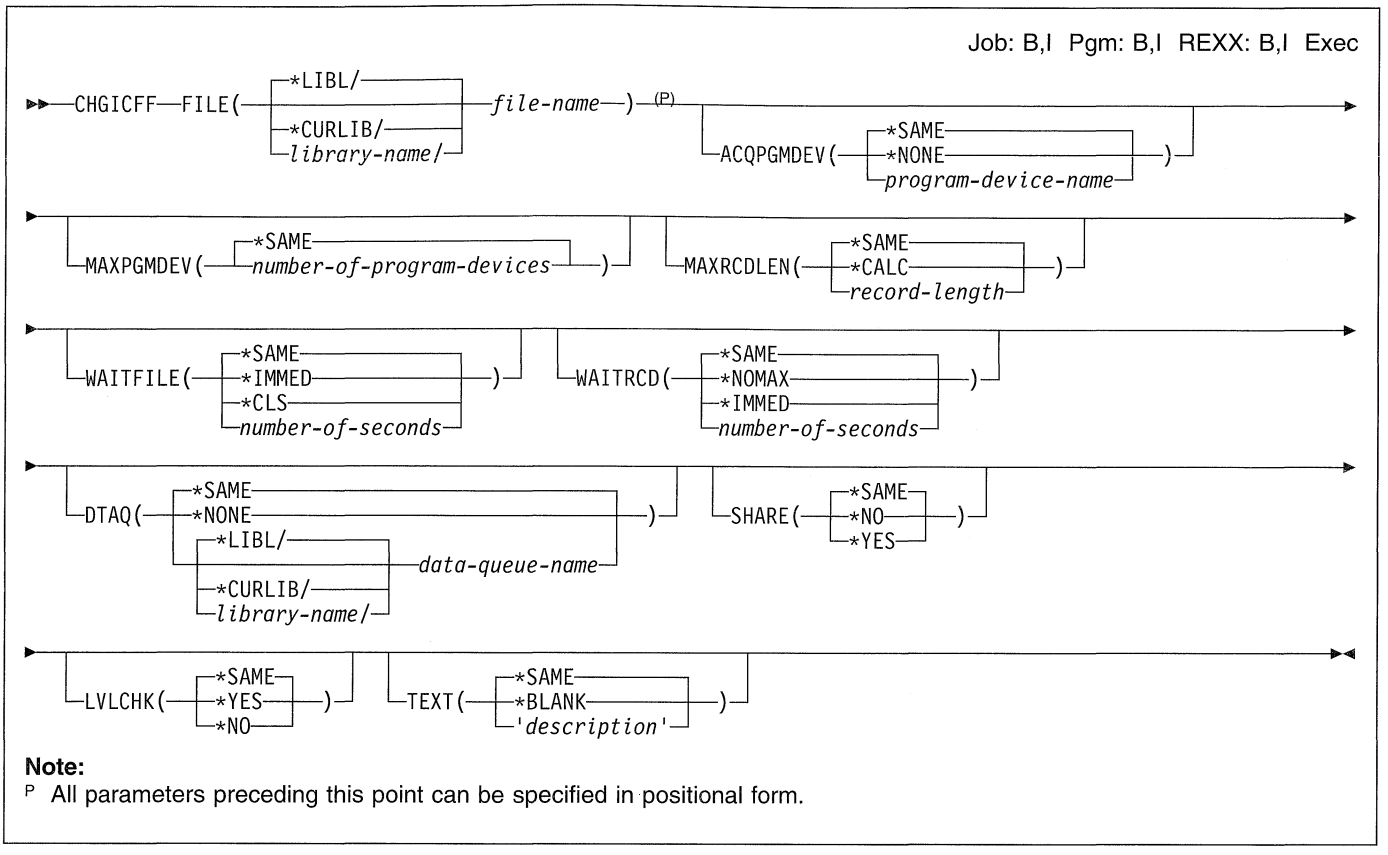
This command changes the program device entry named APPC1 in ICF file ICF1 to have the remote location name *REQUESTER. This program device entry is changed to the FMTSLT(*RMTFMT) and CNVTYPE(*SYS) attributes.

Example 3: Changing the Communications Device

```
CHGICFDEVE  FILE(ICFLIB/ICFAPPL1)  PGMDEV(APPC)
  RMTLOCNAME(APPCMPLS)  DEV(MPLSLINE2)
```

This command changes the program device entry named APPC in file ICFAPPL1. The remote location name is changed to APPCMPLS and the device is changed to MPLSLINE2.

CHGICFF (Change Intersystem Communications Function File) Command



Note:
P All parameters preceding this point can be specified in positional form.

Purpose

The Change Intersystem Communications Function File (CHGICFF) command changes attributes in the file description of an ICF file.

Required Parameter

FILE

Specifies the qualified name of the ICF file whose description is being changed.

The name of the ICF file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

file-name: Specify the name of the file whose description is being changed.

Note: The library and file name have up to 10 characters.

Optional Parameters

ACQPGMDEV

Specifies which program device is acquired when the file is opened.

***SAME:** The value does not change.

***NONE:** The file is opened without a program device being acquired. All program devices used with this file must be explicitly acquired before input/output operations can be started with them.

program-device-name: Specify the name of the first program device acquired when the file is opened. The program device is defined to the file when the file is opened. The name must be specified on the PGMDEV parameter on an Add Intersystem Communications Function Program Device Entry (ADDICFDEVE) command, Change Intersystem Communications Function Program Device Entry (CHGICFDEVE) command, or Override Intersystem Communications Function Program Device Entry (OVRICFDEVE) command before the file is opened.

MAXPGMDEV

Specifies the maximum number of program devices that can be used with the ICF file. The program devices are defined using the ADDICFDEVE, CHGICFDEVE, or

OVRICFDEVE command. The number also determines the maximum number of program devices that can be added to the ICF file by the ADDICFDEVE command.

***SAME:** The value does not change.

number-of-program-devices: Specify a value, ranging from 1 through 256, that specifies the maximum number of program devices that can be added to this ICF file.

MAXRCDLEN

Specifies the maximum record length (bytes) used when the file is opened.

***SAME:** The value does not change.

***CALC:** The length calculated for the largest record in the file is used when the file is opened.

record-length: Specify the maximum record length (in bytes) used when the file is opened. Valid values range from 1 through 32767.

WAITFILE

Specifies the number of seconds that the program waits for the file resources and session resources to be allocated when the file is opened, or for the device or session resources to be allocated when an acquire operation is performed to the file. If those resources are not allocated within the specified wait time, an error message is sent to the program. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

Note: An immediate allocation of the device by the device resource is required when an acquire operation is performed to the file.

***SAME:** The value does not change.

***IMMED:** The program does not wait; when the file is opened, an immediate allocation of the file resources is required.

***CLS:** The job default wait time is used as the wait time for the file resources being allocated.

number-of-seconds: Specify the number of seconds, ranging from 1 through 32767, that the program waits for resource allocation.

WAITRCD

Specifies the number of seconds the program waits for the completion of a read-from-invited-device operation to a multiple device file in a high-level language program. Refer to the appropriate high-level language reference manual to determine when a file is treated as a multiple device file. The program performing the read operation waits for input from all invited devices currently accessing the file. If a record is not returned from an invited device in the specified amount of time, a notify message is sent to the program. This parameter has no effect on an input operation directed to a specific device.

Note: This parameter is also used to specify the time (seconds) that a CL program waits to complete a WAIT command. If a record is not returned from any of the devices that should return a record, an escape message is sent to the CL program. More information on the WAITRCD parameter is in the Receive File (RCVF), Send File (SNDF), Send/Receive File (SNDRCVF), and WAIT (Wait) command descriptions.

***SAME:** The value does not change.

***NOMAX:** There is no limit on the time the system waits for the completion of the operation.

***IMMED:** The program does not wait for the read-from-invited-device operation for the completion of the file. A record must be available from an invited program device when the read-from-invited-program-device operation is performed. If a record is not already available when the read-from-invited-program-device operation is performed, a notify message is sent to the program.

number-of-seconds: Specify the number of seconds, ranging from 1 through 32767 seconds, that the program waits for the completion of the read-from-invited-program-devices.

DTAQ

Specifies the name of the data queue that receives an entry from the system when a data-available event is signaled from an invited display device. The data queue need not exist when the display file is created since the name specified on this parameter is not evaluated until the file is used. More information on the data queue function is in the *CL Programmer's Guide*.

***SAME:** The value does not change.

***NONE:** A data queue does not receive an entry from the system.

The name of the data queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

data-queue-name: Specify the name of the data queue that receives an entry from the system when the data-available event is signaled.

SHARE

Specifies whether the open data path (ODP) for the intersystem communications function file is shared with other programs in the routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

More information on shared database files is in the *Database Guide*.

***SAME:** The value does not change.

***NO:** The ODP created by the program with this attribute is not shared with other programs in the routing step. Every time a program opens the file with this attribute, a new ODP to the file is created and activated.

***YES:** The ODP created with this attribute is shared with each program in the routing step that also specifies SHARE(*YES) when it opens the file.

Note: When SHARE(*YES) is specified and control is passed to a program, a read operation in that program retrieves the next input record. A write operation produces the next output record.

LVLCHK

Specifies whether the record format level identifiers in the program are checked against those in the device file when the file is opened. If so, the record format identifiers in the program must match those in the device file. Because the same record format name can exist in more than one file, each record format is given an internal system identifier when it is created.

***SAME:** The value does not change.

***YES:** The format-level identifiers of the record formats are checked when the file is opened. If the format-level identifiers are not all the same, or if they have not been specified in the program, an open error message is sent to the program that tried to open the file.

***NO:** The level identifiers are not checked when the file is opened.

TEXT

Specifies text that briefly describes the file. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

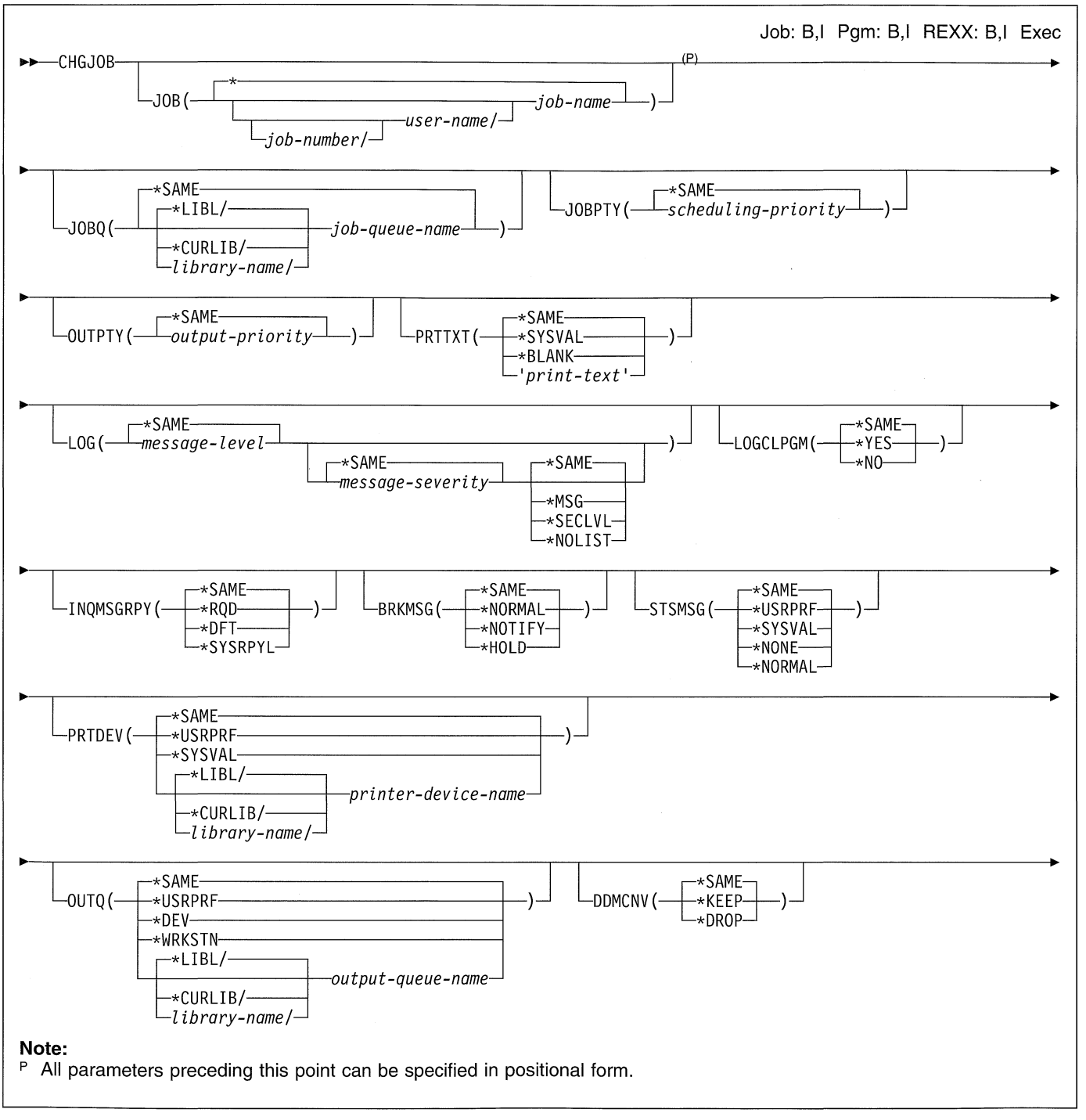
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGICFF   FILE(ICFHIST)  MAXPGMDEV(10)
```

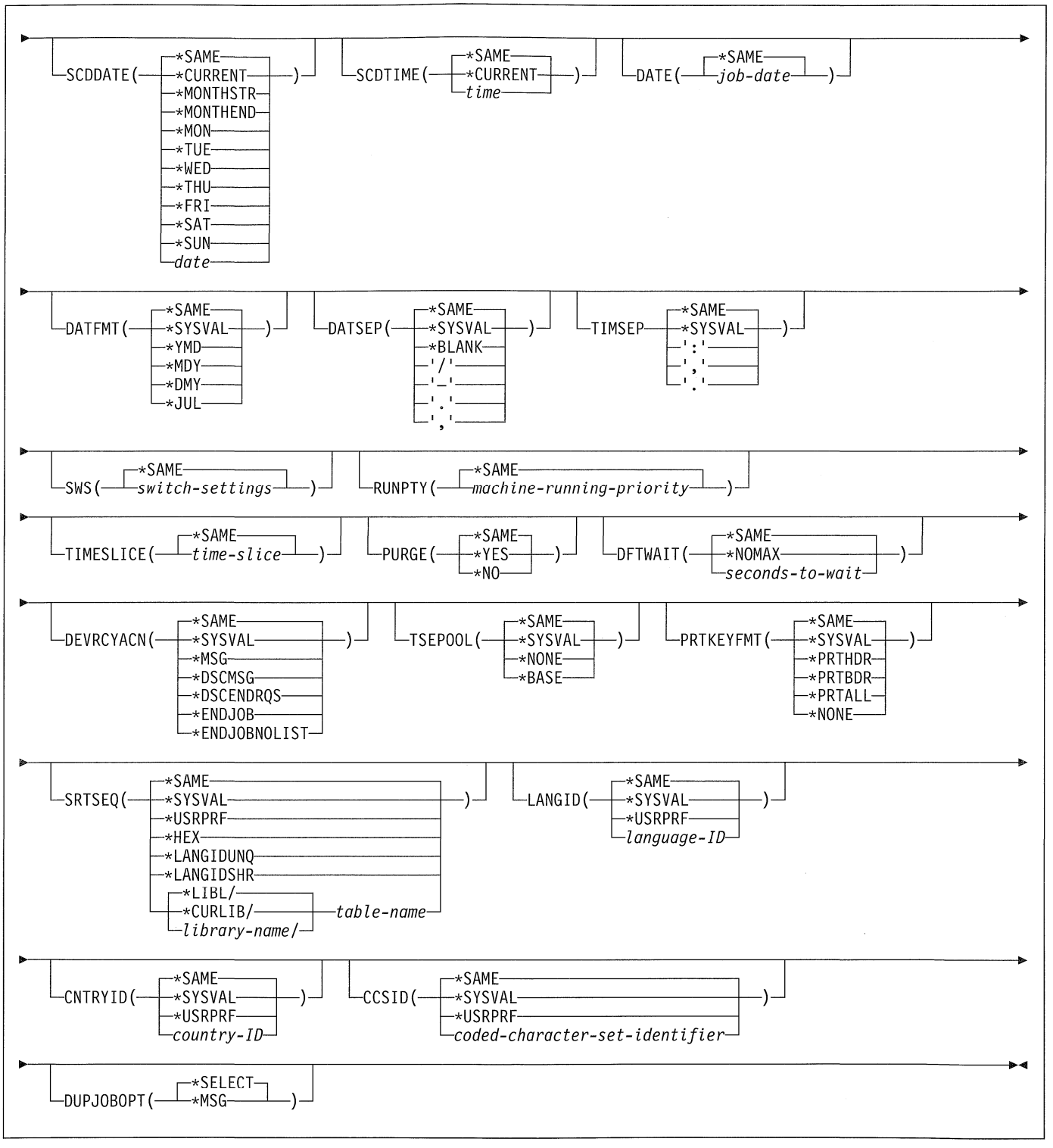
This command changes the maximum number of program devices that can be added to the ICF file ICFHIST.

CHGJOB (Change Job) Command



Note:

P All parameters preceding this point can be specified in positional form.



Purpose

The Change Job (CHGJOB) command changes some of the attributes of a job, including:

- Job queue
- Priorities
- Print text
- Message logging controls
- Logging of CL commands that run
- Inquiry message reply control
- Date format handling
- Date separators
- Printer device
- Output queue
- Distributed Data Management (DDM) conversation setting
- Job switch settings

- Inquiry message reply control
- Break message handling
- Priority of running
- Time slice
- Purge
- Default wait time-out value

With the exception of a change in job queue, the job can be on a job or output queue, or it can be active in a subsystem. The job queue can only be changed when the job is a batch job and is on a job queue. If the job is on the output queue, the only thing that can be changed is the OUTQ priority. With the exception of the class attributes, the new attributes remain in effect for the duration of the job unless they are changed by another CHGJOB command. The class attributes remain in effect until the routing step that is currently active ends. If no routing step is active when the changes are made, they apply to the next routing step that is started. If an attribute that no longer affects the job is changed, a message is sent to the user of the command. For example, if the job has already completed running, it is too late to change the OUTQ and JOBPTY parameters. If any output files are still on the output queue, however, changing the OUTPTY parameter changes their output priority.

Restrictions:

1. The job being changed must use the same user profile as the user issuing the command, or the issuer must have special job control authority (*JOBCTL).
2. Only a user with special job control authority (*JOBCTL) can change the RUNPTY, TIMESLICE, PURGE, or DFTWAIT values.
3. To change JOBQ or OUTQ, the user changing the job must have authority to the new queue. This authority cannot come from program adoption.

Optional Parameters

JOB

Specifies the qualified name of the job and consists of as many as three elements. For example:

job-name
user-name/job-name
job-number/user-name/job-name

*N may be used in place of the user-name element to maintain position in the sequence. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***:** This command is issued within the job whose attributes are changed.

job-name: Specify the name of the job whose attributes are being changed. If no job qualifier is given, all of the jobs currently in the system are searched for the simple job name. If duplicates of the specified name are found, a qualified job name must be specified.

user-name: Specify the name of the user of the job whose attributes are being changed.

job-number: Specify the number of the job whose attributes are being changed.

JOBQ

Specifies the qualified name of the job queue on which this job is placed.

***SAME:** The value does not change.

The name of the job queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

job-queue-name: Specify the qualified name of the job queue on which the submitted job is placed.

JOBPTY

Specifies the scheduling priority used for the job being changed. Valid values range from 0 through 9, where 0 is the highest priority and 9 is the lowest priority. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

scheduling-priority: Specify a value, ranging from 0 through 9, for the job's scheduling priority. If the job is currently on the job queue, its position on the queue in relation to other jobs may be changed. The scheduling priority specified cannot be higher than the priority specified in the user profile under which the job, in which this command is entered, is running.

OUTPTY

Specifies the priority that the job's spooled files have for producing output. Valid values range from 1 through 9, where highest priority is 1 and the lowest priority is 9. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

output-priority: Specify a value, ranging from 1 through 9, for the priority of the job's output files. The output priority specified cannot be higher than the priority specified in the user profile in which the job entering the command is running.

PRTTXT

Specifies up to 30 characters of text to be printed at the bottom of each page of output. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***SYSVAL:** The system value, QPRTTXT, is used.

***BLANK:** Text is not specified.

'print-text': Specify the character string that is printed at the bottom of each page. Up to 30 characters can be entered (enclosed in apostrophes).

LOG

Specifies the message logging values used to determine the amount and type of information sent to the job log by this job. This parameter has three elements: the message (or logging) level, the message severity, and the level of message text. If no values are specified on this parameter, the values specified in the job description associated with this job are used.

Note: For an interactive job, the values specified for the LOG parameter on the Sign Off (SIGNOFF) command take precedence over those specified on the LOG parameter on the Change Job (CHGJOB) command.

The first of the three values in this parameter specifies one of five logging levels.

Element 1: Message Level

The first of the three values in this parameter specifies one of five logging levels.

***SAME:** The value does not change.

message-level: Specify a value, ranging from 0 through 4, that specifies the message logging level used for the job's messages. For additional information on the message levels, refer to the *message-level* value on the LOG parameter of the Create Job Description (CRTJOBD) command description.

Element 2: Message Severity

The second of the three values in this parameter specifies the minimum severity level that causes error messages to be logged in the job log.

***SAME:** The value does not change.

message-severity: Specify a value, ranging from 00 through 99, for the lowest severity level that causes an error message to be placed in the job's log. Only messages with a severity level greater than or equal to this value are logged in the job's log. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

Element 3: Message Text Level

The third of the three values in the LOG parameter specifies the level of message text that is written in the job log.

***SAME:** The value does not change.

***MSG:** Only message text is written to the job's log.

***SECLVL:** Second-level message text is written to the job log.

***NOLIST:** No job log is produced if the job ends normally. If the job ends abnormally (if the end-of-job code is 20 or higher), a job log is produced. The messages appearing in the job log contain message text and help text.

LOGCLPGM

Specifies whether the commands that are run in a control language (CL) program are logged to the job log by way of the CL program's message queue. This parameter sets the status of the job's logging flag. If LOG(*JOB) value has been specified in the Create CL Program (CRTCLPGM) command, the flag set in the LOGCLPGM parameter is used. Other values for the LOG parameter override the LOGCLPGM parameter. The commands are logged in the same manner as requests are logged.

For more information on request logging, refer to the LOG parameter in the CRTCLPGM command description.

***SAME:** The value does not change.

***YES:** Commands that come later in a CL program, which are capable of being logged, are logged to the job log.

***NO:** Commands that come later in a CL program are not logged to the job log.

INQMSGRPY

Specifies the way that inquiry messages, which occur as a result of running a job, are answered. The user can specify that no change is made in the way replies to inquiry messages are sent, that all inquiry messages require a reply by the receiver of the inquiry message, that a default reply be issued, or that the system reply list is checked for a matching reply list entry. The conditions to be met are listed in the system reply list entries; refer to the ADDRPYLE (Add Reply List Entry) command description for more information.

***SAME:** The value does not change.

***RQD:** A reply is required from the receiver of the inquiry messages that are sent during the running of this job.

***DFT:** The default reply to the inquiry message is sent. If no default reply is specified in the message description of the inquiry message, the system default reply, *N, is used.

***SYSRPYL:** The system reply list is checked to see whether there is an entry for inquiry messages issued during the running of this job. If a match occurs, the reply value in that entry is used. If no entry exists for that inquiry message, a reply is required.

BRKMSG

Specifies how break messages are handled for all message queues allocated in *BREAK or *NOTIFY mode for the job.

***SAME:** The value does not change.

***NORMAL:** Break message handling is determined by the message queue status. Messages from the Send Break Message (SNDBRKMSG) command are shown at work stations to which a user is signed on.

***NOTIFY:** An audible alarm sounds to indicate the presence of a message. *NOTIFY is allowed only for interactive jobs. An alarm sounds if:

- The job was previously in BRKMSG(*HOLD) and there are messages on the queue that caused the alarm to sound
- Messages arrive from SNDBRKMSG, or messages arrive that exceed the severity of a message queue in *NOTIFY or *BREAK delivery mode; the messages are not shown

***HOLD:** Break messages are not shown either for message queues in *BREAK delivery mode or for messages sent to a signed on work station through the SNDBRKMSG command. The alarm does not sound for messages sent to message queues in *NOTIFY delivery mode. The user break message handling program is not started.

STSMSG
Specifies whether messages are shown on the display for this job.

***SAME:** The value does not change.

***USRPRF:** The value specified in the user profile is used.

***SYSVAL:** The value in the QSTSMSG system value is used.

***NONE:** Status messages are not shown on the display.

***NORMAL:** Status messages are shown on the display. Text on the bottom line of the display is lost.

PRTDEV
Specifies the qualified name of the default printer device for this job. If OUTQ(*DEV) is specified, the file is placed on an output queue with the same name as the printer.

***SAME:** The value does not change.

***USRPRF:** The printer device name specified in the job's user profile when this command is run is used.

***SYSVAL:** The value specified in the system value QPRTDEV is used.

***WRKSTN:** The output queue assigned to the user's work station is used.

The name of the printer device can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

printer-device-name: Specify the name of the printer device used for this job.

OUTQ
Specifies the qualified name of the output queue used for spooled printer files that specify OUTQ(*JOB). This change does not affect files already created in active jobs or files in completed jobs in which the files were spooled.

***SAME:** The value does not change.

***USRPRF:** The output queue specified in the user profile under which the submitted job runs is used. The user profile is specified on the USER parameter.

***DEV:** The output queue specified on the PRTDEV parameter is used.

***WRKSTN:** The output queue assigned to the user's work station is used.

The name of the output queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

output-queue-name: Specify the name of the output queue.

DDMCNV
Specifies whether distributed data management (DDM) conversations remain active after the completion of each open requested by a source program. Conversations are automatically dropped at the end of the routing step, even when DDMCNV(*KEEP) is specified.

***SAME:** The value does not change.

***KEEP:** The system keeps DDM conversations open when there are no users, except when:

- The routing step ends on the source system either at the job or when the job starts a reroute of the job.
- There is an explicit request that conversations be dropped, using the Reclaim Distributed Data Management Conversation (RCLDDMCNV) command or the Reclaim Resources (RCLRSC) command.
- There is a communications failure or internal systems failure.

***DROP:** The system disconnects a DDM-allocated conversation when there are no users, such as when a single file is using a DDM conversation and the file is closed.

Note: Regardless of the value of DDMCNV, the RCLDDMCNV and RCLRSC commands drop a DDM conversation if there are no users.

SCDDATE

Specifies the date on which the submitted job becomes eligible to run.

If your system or your job is configured to use the Julian date format, the *MONTHSTR and *MONTHEND values are calculated as if the system or job did not use the Julian date format.

***SAME:** The value does not change.

***CURRENT:** The submitted job becomes eligible to run on the current date.

***MONTHSTR:** The submitted job becomes eligible to run on the first day of the month. If you specify *MONTHSTR, and if today is the first day of the month, and if the time you specify on the SCDTIME parameter has not passed, job is eligible to run today. Otherwise, the job becomes eligible to run on the first day of the next month.

***MONTHEND:** The submitted job becomes eligible to run on the last day of the month. If you specify *MONTHEND, and if today is the last day of the month, and if the time you specify on the SCDTIME parameter has not passed, job is eligible to run today. Otherwise, the job becomes eligible to run on the last day of the next month.

***MON:** The job becomes eligible to run on Monday.

***TUE:** The job becomes eligible to run on Tuesday.

***WED:** The job becomes eligible to run on Wednesday.

***THU:** The job becomes eligible to run on Thursday.

***FRI:** The job becomes eligible to run on Friday.

***SAT:** The job becomes eligible to run on Saturday.

***SUN:** The job becomes eligible to run on Sunday.

date: Specify a date in the job date format with or without separators.

SCDTIME

Specifies the time on the scheduled date at which the job becomes eligible to run.

Note: Although the time can be specified to the second, the activity involved in submitting a job and the load on the system may affect the exact time at which the job becomes eligible to run.

***SAME:** The value does not change.

***CURRENT:** The current time is used.

time: Specify a time in the system format with or without separators defined for the job.

DATE

Specifies the date that is assigned to the job. More information on possible date formats is in the *CL Programmer's Guide*.

***SAME:** The value does not change.

job-date: Specify the job date for the job. If no job date is specified for a job, the system date is used as the default for any function requiring a job date. The date specified in this parameter overrides the system date for the running of the job only.

DATFMT

Specifies the date format used for the job.

***SAME:** The value does not change.

***SYSVAL:** The value specified in the system value QDATFMT is used.

***YMD:** The date format used is year, month, and day.

***MDY:** The date format used is month, day, and year.

***DMY:** The date format used is day, month, year.

***JUL:** The Julian date format is used.

DATSEP

Specifies the date separator used for the job.

***SAME:** The value does not change.

***SYSVAL:** The value specified in the system value QDATSEP is used.

***BLANK:** A blank is used for the date separator.

'/': A slash (/) is used for the date separator.

'–': A dash (–) is used for the date separator.

'.': A period (.) is used for the date separator.

',': A comma (,) is used for the date separator.

TIMSEP

Specifies the time separator used for the job.

Note: On other CL commands, time can be specified with or without a time separator. If a time separator is used on another command, it must be the same as the time separator specified on this parameter. If a different time separator is used to specify time on another command, the command will fail.

***SAME:** The time separator does not change.

***SYSVAL:** The value specified in the system value QTIMSEP is used.

':': A colon (:) is used for the time separator.

',': A comma (,) is used for the time separator.

'.': A period (.) is used for the time separator.

***BLANK:** A blank is used for the time separator.

SWS

Specifies the switch settings for a group of eight job switches used with the job. These switches can be set or tested in a CL program and used to control the flow of the program. For example, if a certain switch is on, another program can be called. The job switches may also be valid in other high-level language (HLL) programs. The only values that are valid for each 1-digit switch are 0 (off), 1 (on), or X. The X indicates that a switch value does not change.

***SAME:** The value does not change.

switch-settings: Specify any combination (within quotation marks or not) of eight zeros, ones, or Xs to change the job switch settings. If a switch value does not change, enter an X in the position representing that switch.

RUNPTY

Specifies the run priority for the routing step. Machine run priority is a value, ranging from 1 (highest priority) through 99 (lowest priority), that represents the importance of the routing step when it competes with other routing steps for machine resources. This value represents the relative (not the absolute) importance of the routing step. For example, a routing step with a run priority of 25 is not twice as important as one with a run priority of 50.

***SAME:** The value does not change.

machine-running-priority: Specify the run priority, ranging from 1 through 99, that the routing step uses.

TIMESLICE

Specifies the maximum amount of processor time (in milliseconds) given to the routing step before other routing steps are given the opportunity to run. The time slice establishes the amount of time needed by the routing step to accomplish a meaningful amount of processing. At the end of the time slice, the routing step can be put in an inactive state so that other routing steps can become active in the storage pool.

***SAME:** The value does not change.

time-slice: Specify the amount of time (in milliseconds) that the routing step has to run when it is given processing time. Valid entries range from 1 through 9999999 (9,999,999 milliseconds or 9999.999 seconds).

PURGE

Specifies whether the job is marked to be moved out of main storage and put into auxiliary storage at the end of a time slice or when entering a long wait, such as waiting for a work station user's response.

***SAME:** The value does not change.

***YES:** The job is moved out of main storage and put into auxiliary storage.

***NO:** The job is not moved out of main storage. However, when some of the main storage it occupies is needed to run other jobs in the same storage pool, pages belonging to this job are moved (eight at a time) to auxiliary storage to accommodate pages needed by other jobs. Then, when this job runs again, its pages are returned to main storage as they are needed.

DFTWAIT

Specifies, for the default maximum wait time, that processing of a job is held until a system instruction that requires a wait, such as a lock instruction, completes running. This default wait time is used when a wait time is not otherwise specified for a given situation. Normally, this is the amount of time the system user is willing to wait for the system before canceling the request.

If the wait time for any instruction is exceeded, an error message can be either shown or automatically handled by a Monitor Message (MONMSG) command.

The wait time specified for this parameter is ignored for read operations to database files; to specify that attribute, use the WAITRCD parameter of the appropriate database command (create, change, or override) for a physical, logical, or database file command.

***SAME:** The value does not change.

***NOMAX:** There is no limit on the time the system waits for the completion of the operation.

seconds-to-wait: Specify a value, ranging from 1 through 9999999 seconds, for the maximum time that the system waits for the system instruction to complete running.

DEVRCYACN

Specifies the work station device recovery action to use for the job when an input/output error is encountered on the *REQUESTER device (the device on which the user signed on the system) for interactive jobs.

Note: This attribute is ignored for non-interactive jobs.

***SAME:** The value does not change.

***SYSVAL:** The system value, QDEVRCYACN, is used as the device recovery action for this job.

***MSG:** The application program requesting the input/output operation receives an error message, which indicates that the input/output operation failed.

***DSCMSG:** The job is automatically disconnected. After the job is reconnected, it receives an error message indicating that an input/output error has occurred and that the device has been recovered. Even though the device has been recovered, the contents of the display prior to the error must be shown again.

***DSCENDRQS:** The job is automatically disconnected. After the job is reconnected, the End Request (ENDRQS) command is issued specifying the processor that made the previous request. If no request processor is available, the ENDRQS command fails, and the message issued when the *DSCMSG value is specified is signaled.

***ENDJOB:** The job is ended with the *IMMED option, and a job log is produced for the job.

***ENDJOBNOLIST:** The job is ended with the *IMMED option. No job log list is produced for the job.

TSEPOOL

Specifies whether interactive jobs are moved to another main storage pool when the time slice end is reached. When a long wait occurs, the job is moved back to the pool in which it was originally running. This may minimize the effect on the interactive response time of other interactive jobs.

***SAME:** The value does not change.

***SYSVAL:** The value specified in the system value QTSEPOOL is used.

***NONE:** The job is not moved to another main storage pool when the time slice end is reached.

***BASE:** The job is moved to the base pool when the time slice end is reached.

PRTKEYFMT
Specifies the information to be printed when the print key is pressed.

***SAME:** The value does not change.

***SYSVAL:** The value specified in the system value QPRTKEYFMT is used.

***PRTHDR:** Header information is included with output from the print key.

***PRTBDR:** Border information is included with output from the print key.

***PRTALL:** Header and border information is included with output from the print key.

***NONE:** Header and border information is not printed.

SRTSEQ
Specifies the sort sequence table to be used for string comparisons for this job.

***SAME:** This value does not change.

***SYSVAL:** The system value QSRTSEQ is used.

***USRPRF:** The sort table specified in the job's user profile is used.

***HEX:** A sort sequence table is not used. The hexadecimal values of the characters are used to determine the sort sequence.

***LANGIDUNQ:** A unique-weight sort table is used.

***LANGIDSHR:** A shared-weight sort table is used.

The name of the sort sequence table can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

table-name: Specify the name of the sort sequence table to be used with this job.

LANGID
Specifies the language identifier to be used when SRTSEQ(*LANGIDUNQ) or SRTSEQ(*LANGIDSHR) is specified.

***SAME:** The value does not change.

***SYSVAL:** The system value QLANGID is used.

***USRPRF:** The language ID specified in the job's user profile is used.

language-ID: Specify the language identifier to be used by the job.

CNTRYID
Specifies the country identifier to be used by the job.

***SAME:** The value does not change.

***SYSVAL:** The system value QCNTRYID is used.

***USRPRF:** The country ID specified in the job's user profile is used.

country-ID: Specify the country identifier to be used by the job.

CCSID
Specifies the coded character set identifier (CCSID) used for the job.

A CCSID is a 16-bit number identifying a specific set of encoding scheme identifiers, character set identifiers, code page identifiers, and additional coding-related information that uniquely identifies the coded graphic representation used.

***SAME:** The CCSID value does not change.

***SYSVAL:** The value specified in the system value QCCSID is used.

***USRPRF:** The CCSID specified in the job's user profile is used.

***HEX:** The CCSID 65535 is used.

coded-character-set-identifier: Specify the CCSID. More information on valid CCSIDs is in the *National Language Support Planning Guide*.

DUPJOBOPT
Specifies the action taken when duplicate jobs are found by this command.

***SELECT:** The selection display is shown when duplicate jobs are found during an interactive session. For other types of sessions, a message is issued.

***MSG:** A message is issued when duplicate jobs are found.

Examples

Example 1: Changing Message Logging Values

```
CHGJOB   JOB(123581/DEPT2/WS1)  LOG(2 40 *SECLVL)
```

This command changes the job WS1, which is associated with the user profile DEPT2, and has the job number 123581. It receives only commands and associated diagnostic messages (help text) if the messages have a severity level greater than or equal to 40. Help text, in addition to the message text, is logged in the job log.

Example 2: Changing Scheduling Priority and Job Switches

```
CHGJOB   JOB(PAYROLL)  JOBPTY(4)  OUTPTY(3)
  SWS(10XXXX00)
```

This command changes the scheduling priority for the job PAYROLL to 4 and the priority of the job's output to 3. Also, four of the eight job switches are changed: switches 1 and 2 are set to 1 and 0, switches 3 through 6 remain the same, and switches 7 and 8 are both set to 0. Because only the simple name of the job is specified, there can be only one job named PAYROLL in the system.

Example 3: Moving a Job to Another Queue

```
CHGJOB   JOB(MYJOB)  JOBQ(MYLIB/MYJOBQ)
  INQMSGRPY(*SYSRPYL)
```

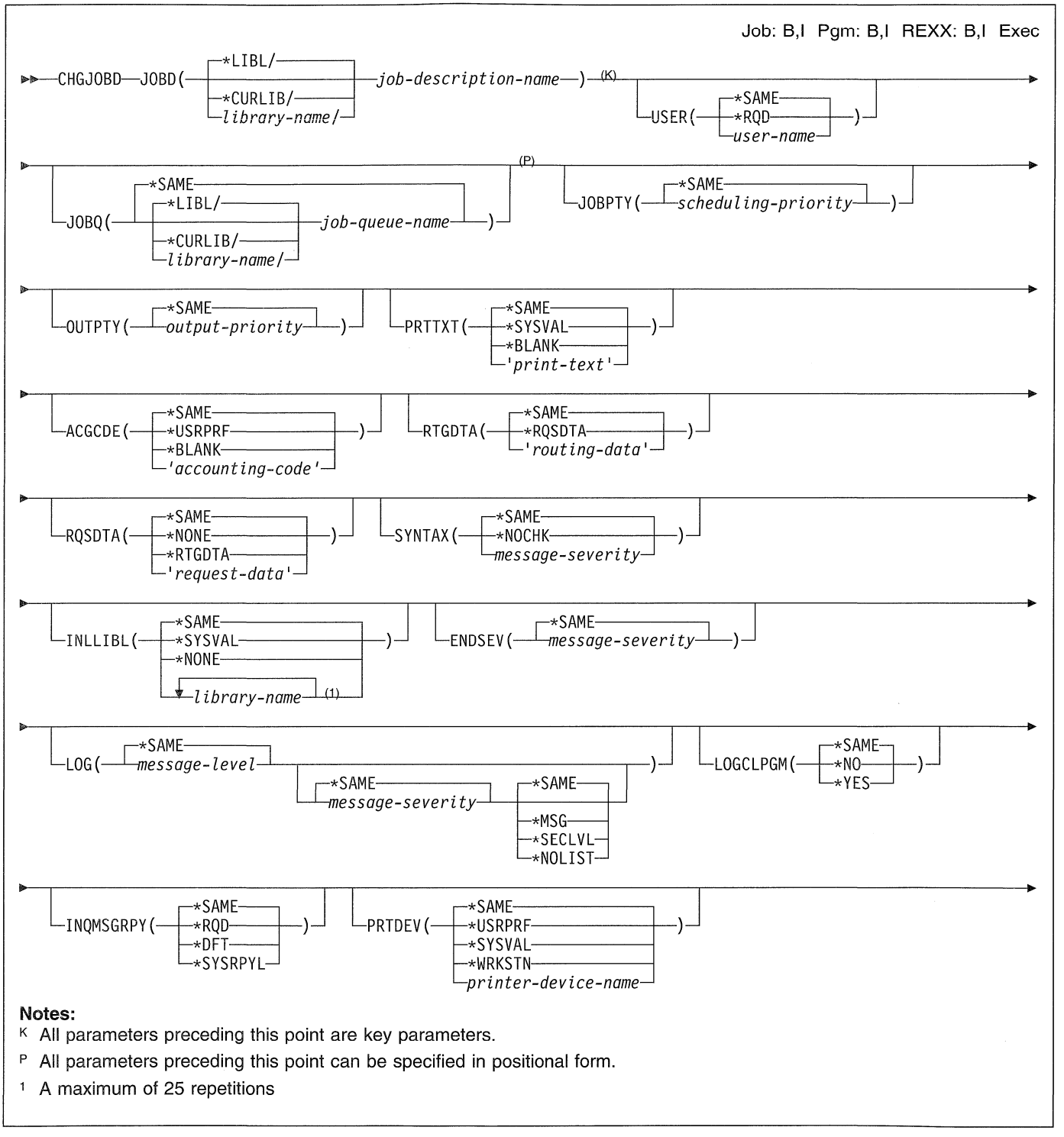
This command moves the job from the job queue it is currently in to the job queue MYJOBQ in library MYLIB. For this change to be made, the job must not have begun running and must be a batch job. Also, the job is changed so that any inquiry message that is issued on behalf of this job that has an entry in the system automatic reply list is answered according to the reply in that system reply list entry. For any inquiry message not represented in the system reply list, a reply is required.

Example 4: Changing Run Priority and Default Wait Time

```
CHGJOB   JOB(MYJOB)  RUNPTY(10)  DFTWAIT(150)
```

This command changes the priority of the running of the job MYJOB to 10 and the default wait time to 150 seconds, if the issuer of the command has job control authority. If the job is active, these changes take effect immediately. If the job is on a job queue, when the next routing step for the job is started, the priority of running and default wait times become these values and not what was specified in the class associated with the routing entry used to start the step.

CHGJOBD (Change Job Description) Command

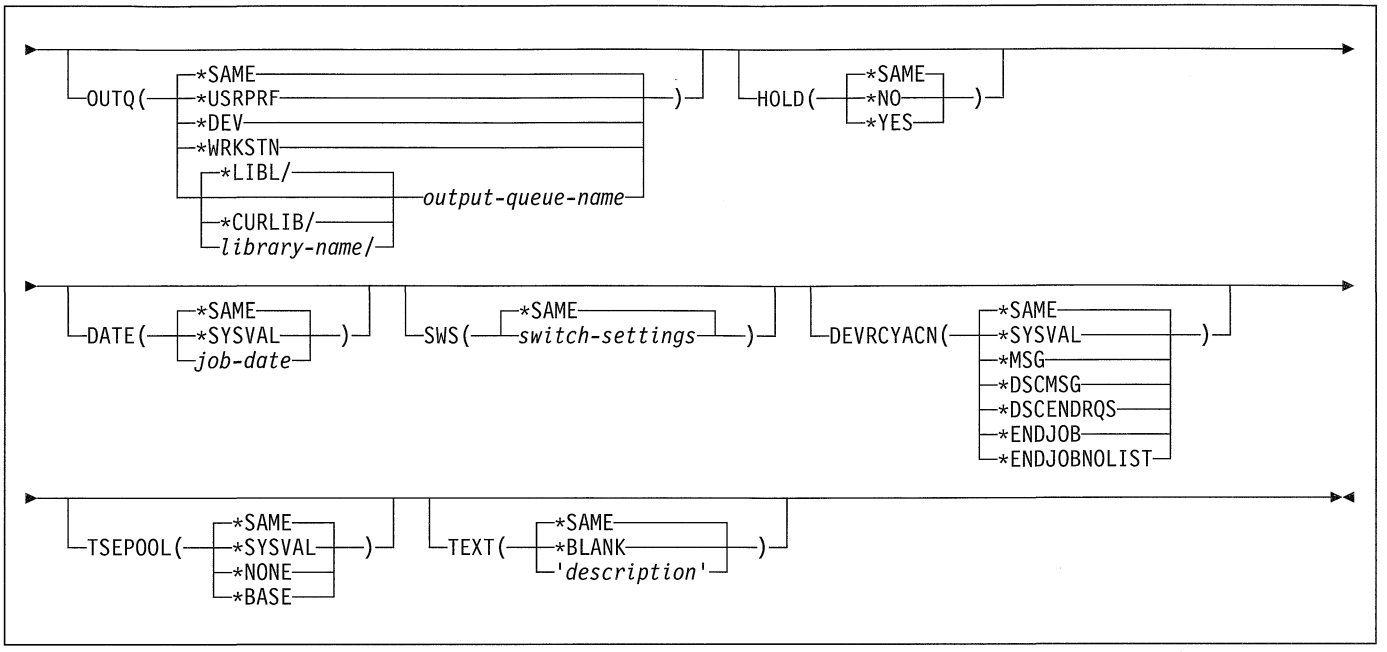


Notes:

K All parameters preceding this point are key parameters.

P All parameters preceding this point can be specified in positional form.

1 A maximum of 25 repetitions



Purpose

The Change Job Description (CHGJOBD) command changes the job-related attributes specified for a job description object through the Create Job Description (CRTJOBD) command. The changes become effective when this command is run.

Any attribute, except the public authority attribute, may be changed. More information on changing object authorizations is in the descriptions of the Revoke Object Authority (RVKOBJAUT) command and Grant Object Authority (GRTOBJAUT) command.

Restrictions:

1. The user of this command must have operational authority for the user profile named in the USER parameter.
2. The user of this command must have object operational and object management authorities for the job description, and read authority for the library in which the job description resides are also required.
3. To change the ACGCDE parameter to a value other than *USRPRF, the user must have *USE authority to the Change Accounting Code (CHGACGCDE) command.

Required Parameter

JOBD

Specifies the qualified name of the job description being changed.

The name of the job description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

job-description-name: Specify the job description name being changed.

Optional Parameters

USER

Specifies the name of the user profile associated with this job description. The names QSECOFR, QSPL, QDOC, QDBSHR, QRJE, QTSTRQS, QDFTOWN, and QSYS are not valid entries for this parameter.

***SAME:** The value does not change.

***RQD:** A user profile name is required to get access to the job description. For work station entries, a password must be entered when signing on the work station; the associated user name becomes the user profile name used for the job. *RQD is not valid for job descriptions specified for job entries that start automatically, or for those used by the Batch Job (BCHJOB) command. It is valid on the Submit Job (SBMJOB) command only if USER(*CURRENT) is specified.

user-name: Specify the user name that identifies the user profile associated with batch jobs using this job description. For interactive jobs, this is the default user name used for signing on the system without typing a password.

JOBQ

Specifies the qualified name of the job queue on which this job is placed.

***SAME:** The name of the job queue does not change.

The name of the job queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

job-queue-name: Specify the qualified name of the job queue associated with this job description.

If the job queue does not exist when the job description is changed, a library qualifier must be specified because the job queue name is retained in the job description.

JOBPTY

Specifies the scheduling priority for jobs that use this job description. Valid values range from 1 through 9, where 1 is the highest priority and 9 is the lowest priority. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

scheduling-priority: Specify a value, ranging from 1 through 9, for the scheduling priority for jobs that use this job description.

OUTPTY

Specifies the output priority of spooled files produced by jobs that use this job description. Valid values range from 1 through 9. The highest priority is 1 and the lowest priority is 9. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

output-priority: Specify a value, ranging from 1 through 9, for the output priority of the spooled files produced by jobs that use this job description.

PRTTXT

Specifies up to 30 characters of text to be printed at the bottom of each page of output. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***SYSVAL:** The system value, QPRTTXT, is used.

***BLANK:** Text is not specified.

'print-text': Specify up to 30 characters, enclosed in apostrophes, to be printed at the bottom of each page.

ACGCDE

Specifies the accounting code used when logging system resource use for a job having this job description. To specify an accounting code other than *USRPRF, the user must have *USE authority for the Change Accounting Code (CHGACGCDE) command.

***SAME:** The value does not change.

***USRPRF:** The accounting code for jobs using this job description is obtained from the user profile for the job.

***BLANK:** An accounting code consisting of 15 blanks is assigned to jobs that use this job description.

'accounting-code': Specify the character string that is used as the accounting code for jobs that use this job description and have accounting statistics logged in the system accounting journal QSYS/QACGJRN. If less than 15 characters are entered, the string is padded with blanks on the right. Note that the character string '*USRPRF' is considered a special value that cannot be used as an accounting code on this command.

RTGDTA

Specifies the routing data used with this job description to start jobs. The routing data determines the routing entry in the subsystem description that identifies the program in which the job runs.

***SAME:** The value does not change.

***RQSDTA:** Up to the first 80 characters of the request data specified in the RQSDTA parameter are used as the routing data for the job.

'routing-data': Specify the character string, enclosed in apostrophes, that is used as the routing data for jobs that use this job description. For example, the value QCMDI is the routing data used by the IBM-supplied interactive subsystem (QINTER and QBASE) to route interactive jobs to the IBM-supplied control language processing unit, QCMD. Up to 80 characters can be specified.

RQSDTA

Specifies the request data placed as the last entry in the job's message queue for jobs using this job description. For example, when a CL command is supplied as request data, it becomes a message that can be read by the control language processing unit, QCMD if the job is routed to QCMD.

***SAME:** The value does not change.

***NONE:** No request data is placed in the job's message queue.

***RTGDTA:** The routing data specified in the RTGDTA parameter is placed as the last entry in the job's message queue.

'request-data': Specify the character string placed as the last entry in the job's message queue as a single request. Up to 256 characters can be entered, enclosed in apostrophes. When a CL command is entered, it must be enclosed in single apostrophes, and where single apostrophes would normally be used *inside* the command, double apostrophes must be used instead.

SYNTAX

Specifies whether requests placed on the job message queue for jobs using this job description are syntax-checked as CL commands. When syntax checking is specified, the commands are checked for syntax as they

are submitted instead of when the job is run. This provides an earlier diagnosis of syntax errors. If checking is specified, the message severity that causes a syntax error to end processing of a job is also specified.

***SAME:** The value does not change.

***NOCHK:** The request data is not checked for syntax as CL commands.

message-severity: Specify a value, ranging from 00 through 99, that specifies the lowest message severity that can cause running of a job to end. The request data is checked for syntax as CL commands; if a syntax error occurs that has severity equal to or greater than the error message severity specified here, the running of the job containing the erroneous command is suppressed. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

If the message severity is specified, it is used only when the job description is used by a job command that also has RQSDTA(*) specified and the requests are CL commands.

INLLIBL

Specifies the initial user part of the library list used for jobs that have this job description.

Note: Duplicating libraries in the library list is not allowed. If specifying a list of libraries for INLLIBL (or if specifying *JOBD and the job description used specifies a list of libraries), ensure that there are no duplicates in the list being used. To do this, compare the libraries in the list to the libraries contained in system value QSYSLIBL. Duplicates must be removed from the INLLIBL parameter.

More information on the use of library lists is in the *CL Programmer's Guide*.

***SAME:** The value does not change.

***SYSVAL:** The system default user library list is used for jobs that use this job description. The default library list contains the library names specified in the system values when a job using this job description is started.

***NONE:** The user part of the initial library list is empty; only the system portion is used.

library-name: Specify the names of one or more libraries in the user part of the library list for jobs that use this job description. No more than 25 names can be specified; the libraries are searched in the same order as they are listed here.

ENDSEV

Specifies the message severity level of escape messages that can cause running of a batch job to end. The batch job is ended when a request in the batch input stream sends to the request processing program an escape message whose severity code is equal to or greater than that specified. This parameter value is compared with the severity of an escape message not monitored that occurs as a result of running a noncompiled CL command in a batch job. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

message-severity: Specify a value ranging from 00 through 50 for the message severity of an escape message that results from a request in the batch input stream and ends the jobs that use this job description. Because escape messages typically have a maximum severity level of 50, a value of 50 or lower must be specified to end a job as a result of an escape message. An unhandled escape message whose severity is equal to or greater than the value specified causes the job to be canceled. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

LOG

Specifies the message logging values used to determine the amount and type of information sent to the job log by this job. This parameter has three elements: the message (or logging) level, the message severity, and the level of message text. If no values are specified on this parameter, the values specified in the job description associated with this job are used.

Note: The first of the three values in the LOG parameter specifies one of five logging levels.

Element 1: Message Level

The first of the three values in the LOG parameter specifies one of five logging levels.

***SAME:** The value does not change.

message-level: Specify a value, ranging from 0 through 4, for the message logging level used for the job's messages. For a description of the message levels, refer to the *message level* variable description in the LOG parameter of the CRTJOBD command.

Element 2: Message Severity

The second of the three values in the LOG parameter specifies the minimum severity level that causes error messages to be logged in the logs of jobs that have this job description.

***SAME:** The value does not change.

message-severity: Specify a value, ranging from 00 through 99, for the lowest severity level that causes an error message to be logged in the job's log. Only messages that have a severity greater than or equal to this value are logged in the job's log. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

Element 3: Message Text Level

The third of the three values in the LOG parameter specifies the level of message text that is written in the job log.

***SAME:** The value does not change.

***MSG:** Only message text is logged in the job's log.

***SECLVL:** Second-level message text is written to the job log.

***NOLIST:** No job log is produced if the job ends normally. If the job ends abnormally (if the end-of-job code is 20 or higher), a job log is produced. The messages appearing in the job log contain message text and help text.

LOGCLPGM

Specifies whether the loggable commands logged and run in a control language program are logged to the job log by way of the CL program's message queue. This parameter sets the status of the job's logging flag. If *NO is specified, the logging flag status is *off* and CL commands are not logged. If *YES is specified and the LOG(*JOB) value is specified in the Create CL Program (CRTCLPGM) command, all commands in the CL program that can be logged are logged to the job log.

For more information on request logging, refer to the LOG parameter in the CRTCLPGM command description.

***SAME:** The value does not change.

***NO:** The commands in a CL program are not logged to the job log.

***YES:** The commands in a CL program are logged to the job log.

INQMSGRPY

Specifies the action taken when a predefined inquiry message is issued by a job that uses this job description. The conditions that must be met are listed in the system reply list entries. Refer to the ADDRPYLE (Add Reply List Entry) command description for more information.

***SAME:** The value does not change.

***RQD:** A reply is required for any predefined inquiry message issued by a job that uses this job description.

***DFT:** The default reply to the inquiry message is sent. If no default reply is specified in the message description of the inquiry message, the system default reply, *N, is used.

***SYSRPYL:** The system reply list is checked to determine whether an entry matches the message identifier and optional compare value for any inquiry message issued by a job that uses this job description. If a match occurs, the reply value in that reply list entry is used. If no entry exists for that message, a reply is required.

PRTDEV

Specifies the qualified name of the default printer device for this job. If OUTQ(*DEV) is specified, the file is placed on an output queue with the same name as the printer.

***SAME:** The value does not change.

***USRPRF:** The name of the printer device for jobs using this job description is taken from the user profile for the job when the job is started.

***SYSVAL:** The value specified in the system value QPRTDEV is used.

***WRKSTN:** The output queue assigned to the user's work station is used.

printer-device-name: Specify the name of the printer device used with this job description.

OUTQ

Specifies the qualified name of the output queue used for spooled printer files that specify OUTQ(*JOB). This change does not affect files already created in active jobs or files in completed jobs in which the files were spooled.

***SAME:** The value does not change.

***USRPRF:** The output queue name for jobs using this job description is taken from the user profile for the job at the time the job is started.

***DEV:** The output queue specified on the PRTDEV parameter is used.

***WRKSTN:** The output queue assigned to the user's work station is used.

The name of the output queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

output-queue-name: Specify the name of the default output queue used by jobs that use this job description.

If the output queue does not exist when the job description is changed, a library qualifier must be specified, because the qualified output queue name is retained in the job description.

HOLD

Specifies whether jobs using this job description are placed on the job queue in the hold condition. A job placed on the job queue in the hold condition is held until it is either released by the Release Job (RLSJOB) command or canceled by the End Job (ENDJOB) or Clear Job Queue (CLRJOBQ) command. If the job is not run before the next power-down of the system, the job queue can be cleared (and the job ended) when the next initial program load (IPL) is done.

***SAME:** The value does not change.

***NO:** Jobs using this job description are not held when they are put on the job queue.

***YES:** The spooled file is held until released by the Release Spool File (RLSSPLF) command.

DATE

Specifies the date that is assigned for jobs that use this job description.

***SAME:** The value does not change.

***SYSVAL:** The value in the QDATE system value at the time the job is started is used as the job date.

job-date: Specify the value used as the job date for the job being started. The format currently specified for the system value QDATFMT must be used. The *Work Management Guide* describes the QDATFMT system value.

SWS

Specifies the initial job switch settings for a group of eight job switches for jobs that use this job description. These switches can be set or tested in a CL program and used to control the flow of the program. For example, if a certain switch is on, another program may be called. The job switches may also be valid in other high-level language (HLL) programs. The only values that are valid for each single-digit switch are 0 (off) or 1 (on).

***SAME:** The value does not change.

switch-settings: Specify any combination of eight 0's or 1's to change the job switch settings.

DEVRCYACN

Specifies the recovery action to take for the job when an input/output error is encountered on the *REQUESTER device (the device on which the user signed on the system) for interactive jobs that use this job description.

Note: This attribute is ignored for non-interactive jobs.

***SAME:** The value does not change.

***SYSVAL:** The system value, QDEVRCYACN, is used as the device recovery action for this job.

***MSG:** The application program requesting the input/output operation receives an error message which indicates that the input/output operation has failed.

***DSCMSG:** The job is automatically disconnected. After the job is reconnected, it receives an error message indicating that an input/output error has occurred and that the device has been recovered. Even though the device has been recovered, the contents of the display when the error occurred must be shown again on the display.

***DSCENDRQS:** The job is automatically disconnected. After the job is reconnected, the ENDRQS command is issued specifying the processor that made the previous request. If a request processor is unavailable, the ENDRQS command fails, and the error message issued during the DSCMSG case is signaled.

***ENDJOB:** The job is ended with the *IMMED option. A job log is produced for the job.

***ENDJOBNOLIST:** The job is ended with the *IMMED option. No job log is produced for the job.

TSEPOOL

Specifies whether interactive jobs are moved to another main storage pool when the end of the time slice is reached. When a long wait occurs, the job is moved back to the pool in which it was originally running. This may help minimize the effect on interactive response time.

***SAME:** The value does not change.

***SYSVAL:** The system value in QTSEPOOL is used as the time slice end pool action for this job description.

***NONE:** The job is not moved when the end of the time slice is reached.

***BASE:** The job is moved to the base pool when the end of the time slice is reached.

TEXT

Specifies text that briefly describes the job description. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Examples

Example 1: Setting Job Running and Output Priorities

```
CHGJOBD  JOBD(QGPL/QPGMR)  JOBPTY(2)  OUTPTY(2)
```

This command allows jobs using the IBM-supplied job description QPGMR in the QGPL library to process with a higher job and output priority than originally specified for QPGMR. QPGMR originally sets job running and output priorities at level 5. More information about the IBM-supplied job description parameter values is in the *CL Programmer's Guide*.

Example 2: Changing Priority Limits

Assume that the user profile was created as follows:

Part 1:

```
CRTUSRPRF  USRPRF(JLRAY)  PASSWORD(GAMMA)
  SPCAUT(*JOBCTL)  PTYLMT(4)  AUT(*NONE)
```

Then attempt to change the priority limits of the job description BATCH5 with the following command:

Part 2:

```
CHGJOBD  JOBD(BATCH5)  USER(JLRAY)  JOBPTY(1)
  OUTPTY(1)
```

Because the priority limit specified in the user profile takes precedence over any limit specified in a job description, an

error message is sent and a priority of 4 is assumed for both job and output priority levels.

Example 3: Requiring a Password to Sign On at Work Station

```
CHGJOBD  JOBD(INT4)  USER(*RQD)  RTGDTA(QCMDI)
  LOG(*SAME)  ACGCDE('USERXYZ CODE123')  TEXT(*BLANK)
```

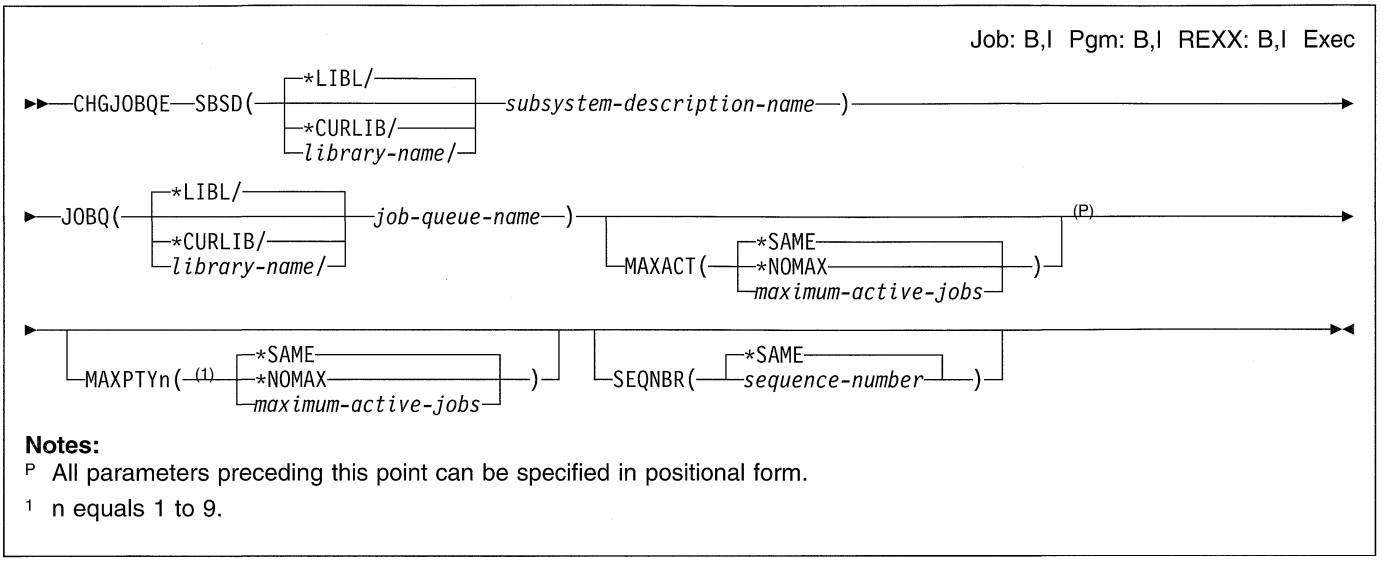
The USER parameter indicates the user of the job description INT4 must enter a password to sign on at a work station. The characters QCMDI are used as the routing data being compared with the routing table of the subsystem under which the job is run. All values of the LOG parameter list are not changed. Jobs that use this job description have accounting data recorded under the accounting code, 'USERXYZ CODE123', when job accounting is active. The text for this job description is changed to all blanks.

Example 4: Directing Jobs Using This System to a Specific Job Queue

```
CHGJOBD   JOBD(BATCH3)  JOBQ(NIGHTQ)  JOBPTY(4)
  OUTPTY(4)  RTGDTA(QCMDB)  INQMSGRPY(*DFT)
  TEXT('Batch #3 JOBD for high priority night work')
```

This command changes a job description named BATCH3 found in the library search list. The jobs using this description are now placed on the job queue NIGHTQ. The priority for jobs using this description and their spooled files is 4. QCMDB is the routing data that is compared with entries in the routing table of the subsystem where the job is run. The text is changed to indicate that this job description is for high priority night work. The INQMSGRPY parameter specifies that all messages of type *INQ that are issued from a job using this job description automatically causes the default reply message to be issued.

CHGJOBQE (Change Job Queue Entry) Command



Notes:

P All parameters preceding this point can be specified in positional form.

1 n equals 1 to 9.

Purpose

The Change Job Queue Entry (CHGJOBQE) command changes an existing job queue entry in the specified subsystem description. This command can be issued while a subsystem is either active or inactive. A job queue entry identifies the job queue from which jobs are selected to run in the subsystem. Jobs can be placed on a job queue by spooling readers or by using the following commands:

- Submit Database Jobs (SBMDBJOB)
- Submit Diskette Jobs (SBMDKTJOB)
- Submit Job (SBMJOB)
- Transfer Job (TFRJOB)
- Transfer Batch Job (TFRBCHJOB)

In a subsystem, job queues with lower sequence numbers are processed first. For more information, refer to the SEQNBR parameter description.

Restriction: To use this command, the user must have object operational and object management authorities for the subsystem description being changed.

Required Parameters

SBSD

Specifies the qualified name of the subsystem description that contains the job queue entry being changed.

The name of the subsystem description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

subsystem-description-name: Specify the name of the subsystem description that contains the job queue entry.

JOBQ

Specifies the qualified name of the job queue that is a source of batch jobs started by the subsystem.

The name of the job queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

If the job queue does not exist when the entry is changed, a library qualifier must be specified because the job queue name is retained in the subsystem description.

job-queue-name: Specify the name of the job queue.

Optional Parameters

MAXACT

Specifies the maximum number of jobs that can be started at the same time from this job queue. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change. However, the maximum activity level of the routing entries might prevent routing steps from being started. If *NOMAX is specified, all the jobs on the job queue are started

(within the limit specified by the MAXJOBS parameter in the subsystem description), even though the activity level of the storage pool being used might prohibit them from running at the same time.

***NOMAX:** There is no maximum number of jobs that can be active at the same time.

maximum-active-jobs: Specify a value for the new maximum for the number of jobs that can be active at the same time from this job queue.

MAXPTYn

Specifies the number of jobs that can be started for a specified job priority level.

***SAME:** The value does not change.

***NOMAX:** There is no disconnect limit.

maximum-active-jobs: Specify the number of jobs started in a specific priority level. Valid values range from 0 through 99. If zero (0) is specified, no jobs are started from a specific priority level.

SEQNBR

Specifies a sequence number for this job queue that is used by the subsystem to determine the order in which the job queues are processed.

***SAME:** The sequence number assigned to this job queue does not change.

sequence-number: Specify the sequence number assigned to the job queue. The sequence number must be unique in the subsystem description. Valid values range from 1 through 9999.

The subsystem first selects jobs from the job queue with the lowest sequence number. When all jobs on that queue have been processed or the number of jobs specified on the MAXACT parameter has been reached, the subsystem processes jobs on the queue with the next higher sequence number. This process continues until all job queue entries have been processed or until the subsystem has reached its limit for overall maximum jobs (as specified on the MAXJOBS parameter in the subsystem description). In some cases, the sequence is interrupted and the subsystem processes a queue with a lower sequence number. This occurs for this subsystem when one of the following conditions occurs:

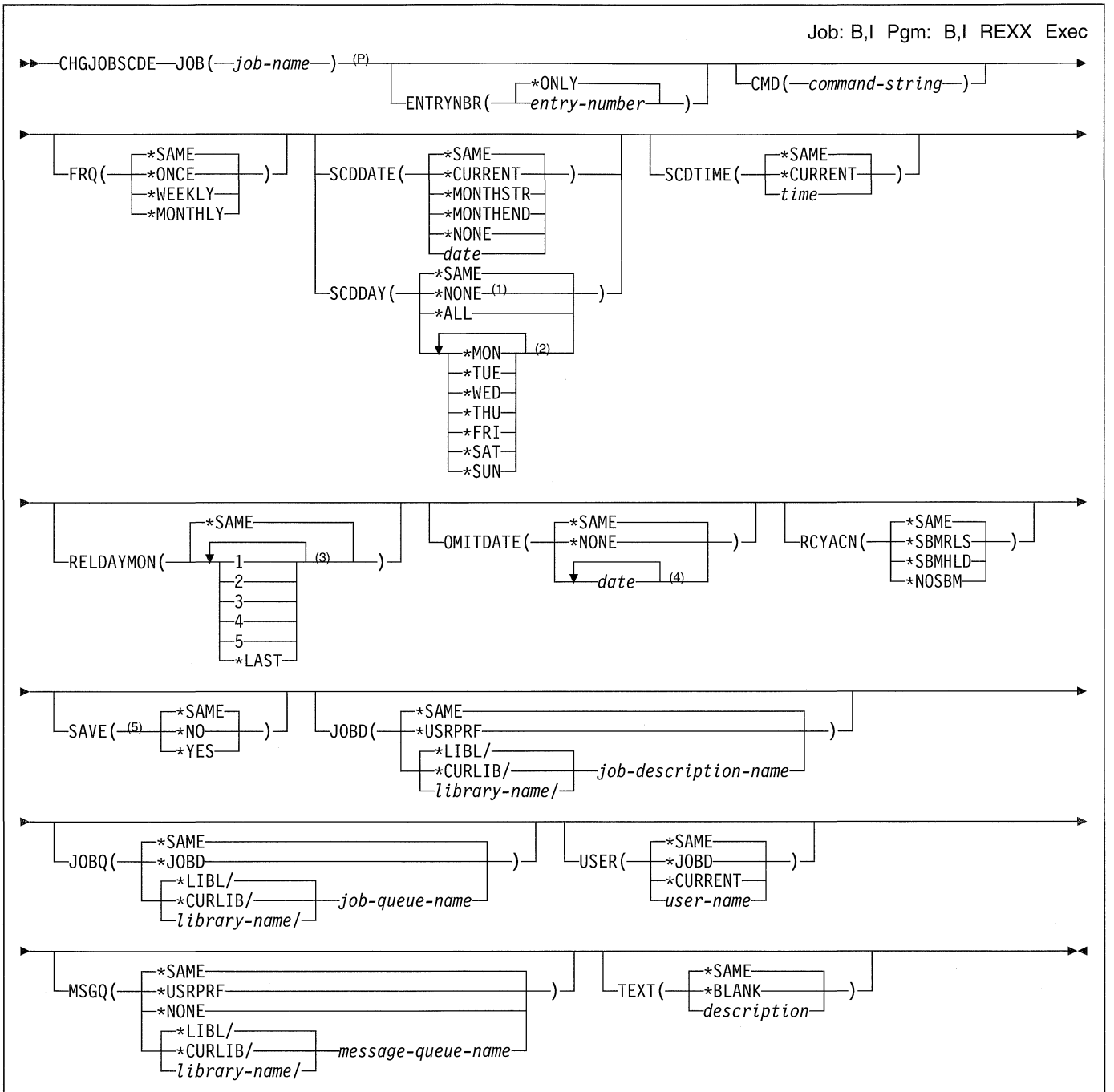
- A held job or job queue is released.
- A job is placed on or transferred to a queue.
- A new queue is allocated.
- A job ends.

Example

```
CHGJOBQE  SBSD(QGPL/QBATCH)  JOBQ(QGPL/QBATCH)
  MAXACT(4)  MAXPTY1(1)  MAXPTY2(*NOMAX)
```

This command changes the maximum number of jobs submitted from the job queue QBATCH via the job queue entry to the QBATCH subsystem for processing at the same time. Up to four jobs from the QBATCH job queue can be active at the same time. The sequence number of the job queue entry does not change. Up to one job can be active from priority level 1. There is no maximum for the number of jobs that can be active at the same time from priority level 2. Priority levels 3 through 9 do not change.

CHGJOBSCDE (Change Job Schedule Entry) Command



Notes:

P All parameters preceding this point can be specified in positional form.

1 SCDDAY(*NONE) is not valid when SCDDATE(*NONE) is specified.

2 A maximum of 7 repetitions

3 A maximum of 5 repetitons

4 A maximum of 20 repetitions

5 SAVE is displayed only when FRQ(*ONCE) is specified.

Purpose

The Change Job Schedule Entry (CHGJOBSCDE) command allows you to change an entry in the job schedule.

The job schedule entry contains all of the information needed to automatically run a batch job. You can use this command to change:

- The command the job runs
- The date and time at which a job is scheduled to run
- The recovery action the system takes if the job cannot be submitted at the scheduled time
- The job description and user profile under which the job is run
- The job queue on which the job is placed when it is submitted
- The message queue to which messages are sent

Each job schedule entry is identified by the job name, which is specified on the JOB parameter of this command, and an entry number, which is assigned by the system when the entry is added. The message replacement text for the message sent when an entry is added contains the entry number. If there is more than one entry with the same job name, you may need to specify the number when changing the entry using this command, removing the entry using the Remove Job Schedule Entry (RMVJOBSCDE) command, or holding or releasing the entry using the Hold Job Schedule Entry (HLDJOBSCDE) or Release Job Schedule Entry (RLSJOBSCDE) command. You can use the Work with Job Schedule Entries (WRKJOBSCDE) command to show or print entries.

More information is in the *Work Management Guide*.

Restrictions:

1. You must have *JOBCTL special authority to use this command; otherwise you can change only those entries that you added.
2. The user must have use authority to the job description and the user profile.
3. The user must have use and add authorities to the message queue.
4. The user must have read authority to the job queue and to all libraries associated with the specified objects.

Required Parameter

JOB

Specifies the name of the job schedule entry.

Optional Parameters

ENTRYNBR

Specifies the number of the job schedule entry you want to change. The message replacement text for the message that is sent when an entry is successfully added contains the entry number. You can also determine the entry number by using the Work with Job Schedule Entries (WRKJOBSCDE) command. Press F11 from the Work with Job Schedule Entries display to show the entry numbers of the selected entries.

***ONLY:** One entry in the job schedule has the job name specified on the JOB parameter. If *ONLY is specified and more than one entry has the specified job name, no entries are changed and a message is sent.

entry-number: Specify the number of the job schedule entry you want to change.

CMD

Specifies the command that runs in the submitted job. The IBM-supplied default routing program QCMD must be used when the job is started or the job will not run. Because the command you specify is used for the request data, the value specified on the RQSDTA parameter in the job description is ignored. The command you specify is syntax-checked when the entry is added.

You can specify a maximum of 512 characters. You can use command prompting for the command you are specifying on this parameter if you are using command prompting for the CHGJOBSCDE command.

Note: If you do not specify a value on this parameter, the command that runs in the submitted batch job does not change.

FRQ

Specifies how often the job is submitted to run.

***SAME:** The value does not change.

***ONCE:** The job is submitted once.

***WEEKLY:** The job is submitted on the same day or days of each week at the scheduled time.

***MONTHLY:** The job is submitted on the same day or days of each month at the scheduled time.

If you specify *MONTHLY and a month does not contain the date specified on the SCDDATE parameter, the job is not run that month. For example, if SCDDATE(01/31/91) and FRQ(*MONTHLY) are specified, the job will run on 01/31, 03/31, 5/31, 7/31, 8/31, 10/31, and 12/31, but will not run in February, April, June, September, or November. To submit a job on the last day of every month, specify SCDDATE(*MONTHEND).

If you specify *MONTHLY and your system or your job is configured to use Julian date format, the job is submitted to run on the day of the month that it would run if the system or job did not use Julian date format.

SCDDATE

Specifies the date on which the job is submitted to run.

If your system or your job is configured to use the Julian date format, the *MONTHSTR and *MONTHEND values are calculated as if the system or job did not use the Julian date format.

***SAME:** The value does not change.

***CURRENT:** The current date is used.

***MONTHSTR:** The job is submitted on the first day of the month. If you specify *MONTHSTR, and if today is the first day of the month, and if the time you specify on the SCDTIME parameter has not passed, the job is submitted today. Otherwise, the job is submitted on the first day of the next month.

***MONTHEND:** The job is submitted on the last day of the month. If you specify *MONTHEND, and if today is the last day of the month, and if the time you specify on the SCDTIME parameter has not passed, the job is submitted today. Otherwise, it is submitted on the last day of the next month.

***NONE:** No date is specified for when a job should be submitted.

date: Specify the date in the job date format.

SCDDAY

Specifies the day of the week on which the job is submitted.

If today is the day of the week specified on this parameter and the time specified on the SCDTIME parameter has not passed, the job is submitted today. Otherwise, the job is submitted on the next occurrence of the specified day. For example, if SCDDAY(*FRI) and SCDTIME(12:00:00) are specified, and you are adding this job schedule entry at 11:00 a.m. on a Friday, the job is submitted today. If you are adding the entry at 4:00 p.m. on a Friday, or at 11 a.m. on a Monday, the job is submitted the following Friday.

***SAME:** The value does not change.

***NONE:** No day is specified for when a job should be submitted.

***ALL:** The job is submitted every day.

***MON:** The job is submitted on Monday.

***TUE:** The job is submitted on Tuesday.

***WED:** The job is submitted on Wednesday.

***THU:** The job is submitted on Thursday.

***FRI:** The job is submitted on Friday.

***SAT:** The job is submitted on Saturday.

***SUN:** The job is submitted on Sunday.

SCDTIME

Specifies the time on the scheduled date at which the job is submitted to run.

Note: Although the time can be specified to the second, the activity involved in submitting a job and the load on the system may affect the exact time at which the job is submitted.

***SAME:** The value does not change.

***CURRENT:** The job is submitted at the current time. If you specify *CURRENT on this parameter and SCDDATE(*CURRENT), the job is immediately submitted to the job queue with a status of released (RLS).

time: Specify the time you want the job to start. The time is specified in 24-hour format with or without a time separator as follows:

- With a time separator, specify a string of 5 or 8 digits where the time separator separates the hours, minutes, and seconds. If this command is entered from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command fails.
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds. Valid values for **hh** range from 00 through 23. Valid values for **mm** and **ss** range from 00 through 59.

RELDAYMON

Specifies the relative day of the month on which the job is submitted to run.

You can specify a value on this parameter only if the SCDDAY parameter and FRQ(*MONTHLY) are specified.

***SAME:** The value does not change.

1: The job is submitted on the specified day of the week the first time it occurs in the month. For example, if you specify SCDDAY(*TUE), FRQ(*MONTHLY), and RELDAYMON(1), the job is submitted on the first Tuesday of every month.

2: The job is submitted on the specified day the second time it occurs in the month.

3: The job is submitted on the specified day the third time it occurs in the month.

4: The job is submitted on the specified day the fourth time it occurs in the month.

5: The job is submitted on the specified day the fifth time it occurs in the month.

***LAST:** The job is submitted on the specified day the last time it occurs in the month.

OMITDATE

Specifies a list of dates on which the job is not submitted. You can, for example, use this parameter to prevent recurring jobs from running on holidays. The date must be specified in the job date format.

***SAME:** The value does not change.

***NONE:** No dates are specified when the job should not be submitted.

date: Specify the dates on which the job is not submitted.

RCYACN

Specifies the recovery action to be taken if the scheduled job cannot be submitted at the designated time

because the system is powered down or in restricted state. The action specified on this parameter occurs at the next initial program load (IPL) or when the system comes out of restricted state.

Jobs submitted during IPL or when the system comes out of restricted state are submitted in the order that they would have been had they been submitted at the times specified in the job schedule entries. If multiple occurrences of a recurring job are missed, the job is submitted only once. The first missed occurrence of a recurring job is used to order the jobs. The next occurrence of the job is calculated from the current date.

Since the scheduler portion of IPL need not be complete for the IPL of the system to be complete, other jobs may start on the system before all of the scheduled jobs have been submitted.

This parameter does not apply in the following instances:

- When a job is released after being held at the date and time it was to be submitted
- When the date and time at which a job is to be submitted passes because of changes to date and time system values

***SAME:** The value does not change.

***SBMRLS:** The job is submitted in the released (RLS) state.

***SBMHLD:** The job is submitted in the held (HLD) state.

***NOSBM:** The job is not submitted.

Specifying *NOSBM affects only missed occurrences of the job. If the job schedule entry is a recurring job, future occurrences are not affected.

SAVE

Specifies whether the entry for a job that is submitted only once is kept after the job is submitted. This parameter is valid only if FRQ(*ONCE) is specified.

***SAME:** The value does not change.

***NO:** The entry is not kept after the job is submitted.

***YES:** The entry is kept after the job is submitted. If you specify *YES, the job is submitted once. The job is not submitted again until the Change Job Schedule Entry (CHGJOBSCDE) command is used to specify a new date and time.

JOBD

Specifies the qualified name of the job description used when submitting the job.

***SAME:** The value does not change.

***USRPRF:** The job description specified in the user profile under which the submitted job runs is used. The user profile is specified on the USER parameter.

The name of the job description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

job-description-name: Specify the name of the job description.

JOBQ

Specifies the qualified name of the job queue on which this job is placed.

You must have authority to the queue to specify a name on this parameter. Authority to the queue cannot be received through program adoption.

***SAME:** The value does not change.

***JOBD:** The submitted job is placed on the job queue specified in the job description. The job description is specified on the JOBD parameter.

The name of the job queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

job-queue-name: Specify the name of the job queue.

USER

Specifies the name of the user profile under which the scheduled job is submitted.

***SAME:** The value does not change.

***CURRENT:** The user profile under which the current job is running is used.

***JOBD:** The user profile specified on the USER parameter of the job description is used. The job description is specified on the JOBD parameter of this command. This value is not valid if USER(*RQD) is specified in the job description.

user-name: Specify the name of the user profile that is used. You must be authorized to the user profile. The user profile must be authorized to the job queue, job description, and message queue specified on this command.

MSGQ

Specifies the qualified name of the message queue to which messages are sent.

Messages are sent when the job is submitted and when a submitted job has completed running. Messages indicating a serious error are sent to the QSYSOPR

message queue regardless of the value specified on this parameter when:

- The message queue specified on this parameter is damaged.
- MSGQ(*NONE) is specified.
- MSGQ(*USRPRF) and USER(*JOBD) are specified, and the job description specified on the JOBD parameter is changed to USER(*RQD) after the entry is added.

Note: When MSGQ(*USRPRF) is specified and the user profile contains a message queue name with *LIBL as its library, the results can be unpredictable. When the job is submitted, the library list from the system value object is used.

***SAME:** The value does not change.

***USRPRF:** The message queue specified in the user profile under which the submitted job runs is used. The user profile is specified on the USER parameter.

***NONE:** Completion messages are not sent. Error messages are sent to the QSYSOPR message queue.

The name of the message queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

message-queue-name: Specify the name of the message queue to which messages are sent.

TEXT

Specifies text that briefly describes the job schedule entry. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Examples

Example 1: Changing a Job Schedule Entry

```
CHGJOBSCDE   JOB(BACKUP)  ENTRYNBR(001584)
  JOBQ(QGPL/QBATCH)
```

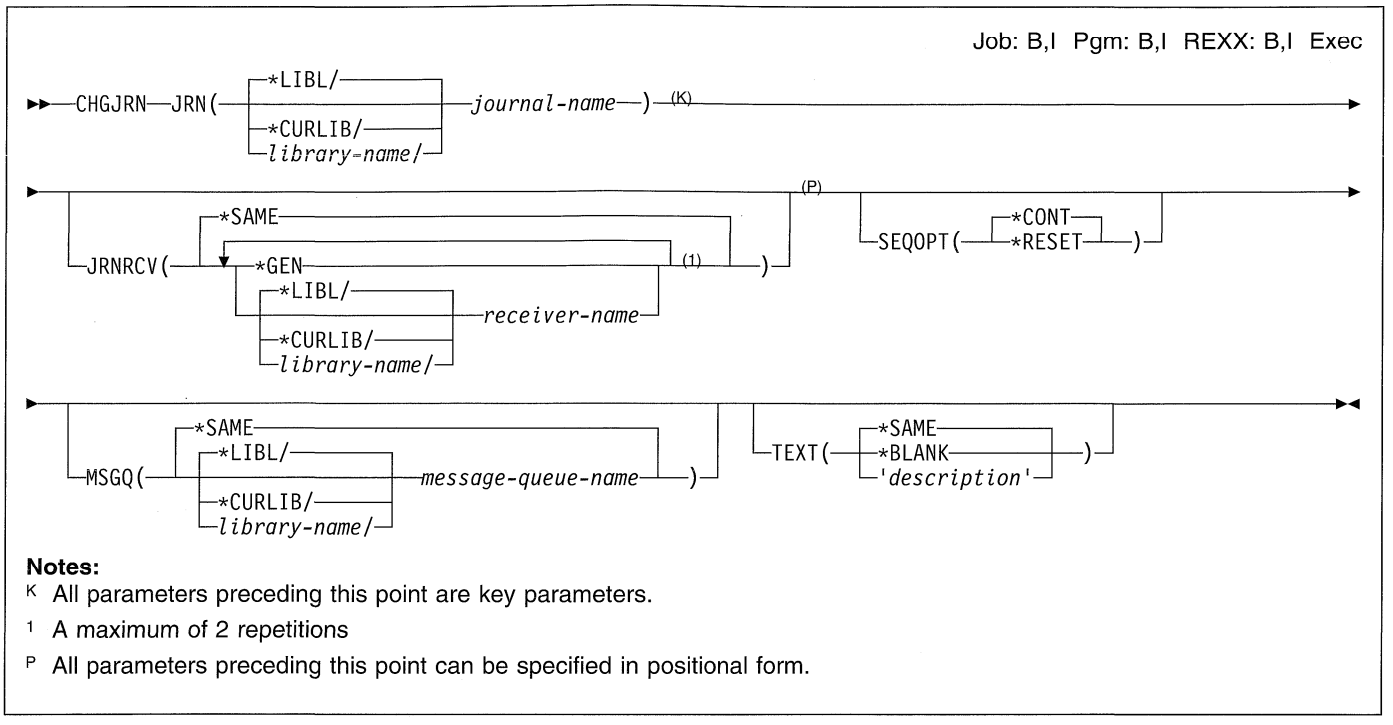
This command changes job schedule entry BACKUP number 001584 so that its jobs are submitted to job queue QBATCH in library QGPL.

Example 2: Changing a Job Schedule Entry

```
CHGJOBSCDE   JOB(EXAMPLE)  ENTRYNBR(*ONLY)
  CMD('CALL PGM(A)')  FRQ(*WEEKLY)
  SCDDATE(121591)  SCDTIME(110000)
```

This command schedules a batch job to run program A at 11 a.m. on 12/15/91 and every week on that same day.

CHGJRN (Change Journal) Command



Notes:

K All parameters preceding this point are key parameters.

1 A maximum of 2 repetitions

P All parameters preceding this point can be specified in positional form.

Purpose

The Change Journal (CHGJRN) command changes the journal receivers, the message queue, or the text associated with the specified journal. The command allows up to two journal receivers to be attached to the specified journal. These replace all previously attached journal receivers. The assigned journal receivers begin receiving journal entries for the journal immediately. The sequence numbering of journal entries can be reset when the receivers are changed. If the sequencing is not reset, an informational message is sent indicating the first sequence number in the newly attached receivers. If the first sequence number is greater than 2,000,000,000, an informational message is sent to the system operator recommending that the sequence numbers be reset.

If JRNRCV is *SAME, the currently attached journal receivers remain attached.

Restrictions:

1. This command is shipped with public *EXCLUDE authority and the QPGMR, QSYSOPR, and QSRV user profiles have private authorities to use the command.
2. Receivers that already contain journal entries cannot be reattached to a journal.
3. No more than two journal receivers can be attached to the journal at any specific time.
4. A message queue that is in library QTEMP cannot be specified for the MSGQ parameter.
5. If changes journaled are applied or removed while this command is running, the user cannot switch journal receivers and JRNRCV(*SAME) must be specified.
6. Resetting of sequence numbers is not valid if JRNRCV is *SAME, or if any files being journaled are open and contain changes that have not yet been forced to auxiliary storage. When the maximum sequence number of 2,147,483,647 is reached, an error message is sent (entry not journaled) and all subsequent operations that require journaling end.

Required Parameter

JRN

Specifies the qualified name of the journal for which journal receivers or operational attributes are being changed.

The name of the journal can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

journal-name: Specify the journal having journal receivers or operational attributes changed.

Optional Parameters

JRNRCV

Specifies which journal receivers to attach to the assigned journal.

***SAME:** The value does not change.

***GEN:** The journal receiver is created by the system and then attached to the assigned journal. The journal receiver is created with the same attributes, with the same owner, with the same authorities, with the same audit level, and in the same library as the currently-attached journal receiver. The name of the new journal receiver is derived by appending a 4-digit number to a portion of the name of the current receiver, or by adding 1 to the number in the name of the current journal receiver. The name of the journal receiver that was created and attached is returned in an informational message. More information on journal receiver names is in the *Advanced Backup and Recovery Guide*.

The name of the journal receiver can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

receiver-name: Specify the names of the journal receivers being attached to the identified journal. The journal receivers must have been previously created in the specified library, and must not have been previously attached to any journal.

Up to 2 journal receivers can be attached at 1 time. Any combination of *GEN and receiver name is valid.

SEQOPT

Specifies whether the journal sequence numbers are continued from the currently attached journal receivers or the journal sequence number is reset to 1 in the newly attached journal receivers.

***CONT:** The journal sequence number of the next journal entry created is 1 greater than the sequence number of the last journal entry in the currently attached journal receivers.

***RESET:** The journal sequence numbers in the newly attached journal receivers are reset to 1. *RESET is not valid if JRNRCV(*SAME) is specified or if any file being journaled is open and contains changes that have not yet been forced to auxiliary storage.

MSGQ

Specifies whether the message queue associated with the journal is changed. The message sent when a journal receiver's storage limit (threshold) is surpassed is sent to this message queue. To set the threshold value, refer to the Create Journal Receiver (CRTJRNRCV) command.

***SAME:** The message queue does not change.

The name of the message queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

message-queue-name: Specify the name of the message queue which replaces the message queue previously specified and to which the message queue is being sent.

TEXT

Specifies text that briefly describes the journal changes. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Examples

Example 1: Detaching Journal Receivers and Resetting Sequence Numbers

```
CHGJRN   JRN(JRNLA)  JRNRCV(RCV10)  SEQOPT(*RESET)
```

This command detaches all journal receivers currently attached to journal JRNLA (JRNLA is found by using the library search list *LIBL). Journal receiver RCV10 (found using the library search list *LIBL) is attached to journal JRNLA. Because SEQOPT(*RESET) is specified, the first journal entry in journal receiver RCV10 has a sequence number of one.

Example 2: Creating and Attaching New Journal Receivers

```
CHGJRN   JRN(JRNLA)  JRNRCV(*GEN *GEN)
```

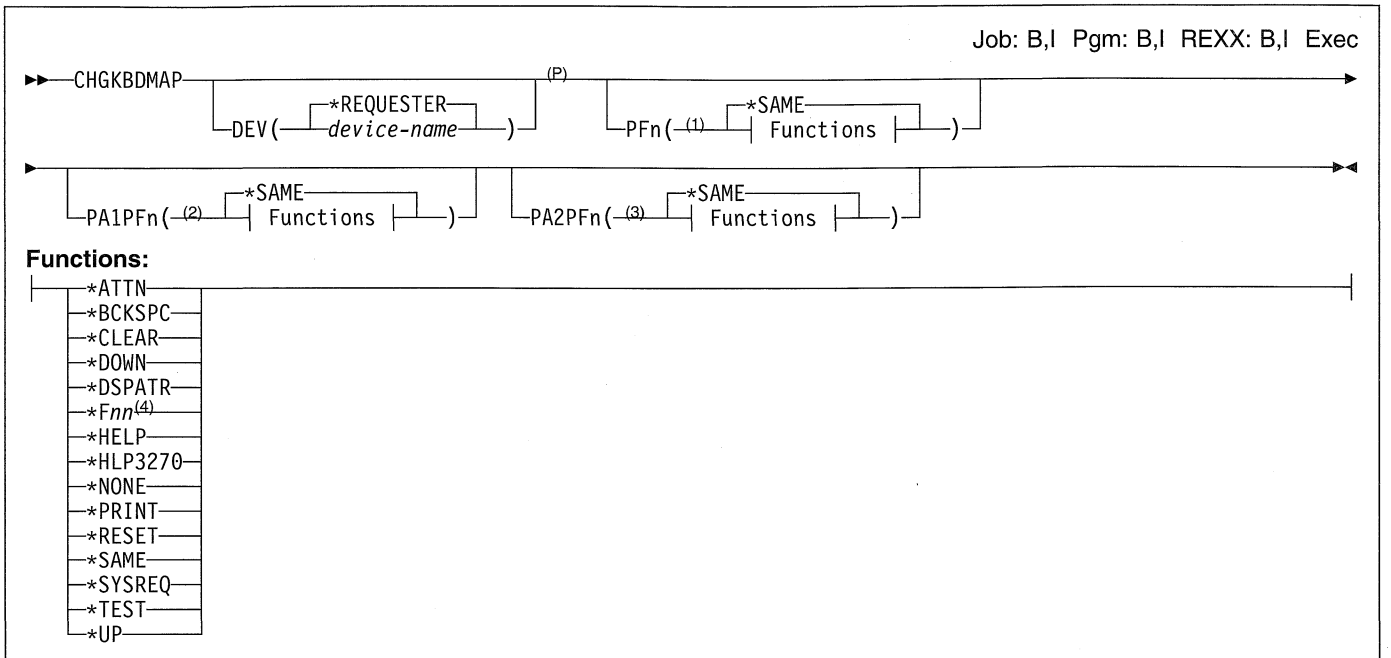
This command detaches all journal receivers currently attached to journal JRNLA. Two new journal receivers are created and attached to journal JRNLA. The libraries, owners, authorities, and auditing levels of the new journal receivers are the same as those of the detached receivers. The names of the new receivers depend on the names of the detached receivers. (For example, if one receiver is named RCVJRNA, the new receiver is named RCVJRN0001. If the receiver is named RCVJRN0001, the new receiver is named RCVJRN0002.) The first journal entry in the new journal receivers has a sequence number one greater than the last sequence number in the detached receivers.

Example 3: Detaching Current Journal Receivers and Creating a New One

```
CHGJRN   JRN(SS/J)  JRNRCV(*N *GEN)
```

This command detaches all journal receivers currently attached to journal J in library SS. One new journal receiver is created and attached to journal J. The name of the new receiver depends on the name of the detached dual receiver, if there is one. Otherwise, it depends on the name of the single detached receiver. For example, if the detached primary and dual receivers are RCVA and RCVB, then the new receiver is named RCVB0001. The new journal receiver has the same library, owner, authorities, and auditing level as the detached dual receiver. The first journal entry in the new journal receiver has a sequence number of one greater than the last sequence number on the detached receivers.

CHGKBDMAP (Change Keyboard Map) Command



Notes:

1 Valid PF values range from PF1 through PF24.

2 Valid PA1PF values range from PA1PF1 through PA1PF12.

3 Valid PA2PF values range from PA2PF1 through PA2PF12.

P All parameters preceding this point can be specified in positional form.

4 Valid *nn* values range from 1 through 24.

Purpose

The Change Keyboard Map (CHGKBDMAP) command allows the user to change the Program Attention (PA) and Program Function (PF) key assignment defaults provided either by the AS/400 system or by a previously specified keyboard mapping. This command assigns the specified F-to-PF map to the device on which the command was entered (if it is a 3270 work station device), or to the 3270 work station specified if the user has authority to that device. More information on user-assignable keyboard mapping is in the *New User's Guide*.

Optional Parameters

DEV

Specifies a valid 3270 display station to which keyboard changes are assigned by using the keyboard mapping command. The user must have allocation authority to the specified device before entering the command. In a program environment, the user should either acquire or allocate the specified device before entering this command.

***REQUESTER:** These PF key changes are assigned to the device on which the command was entered.

device-name: Specify a device other than the device on which the command was entered (*REQUESTER).

KEY SEQUENCE PARAMETERS

Specifies the function assigned to the keys or key sequences ranging from PF1 through PF24, PA1PF1 through PA1PF12, and PA2PF1 through PA2PF12.

***SAME:** The PA and PF key assignments do not change.

function: Specify the function assigned to the PA and PF keys or key sequences.

Each valid key or key sequence is a separate parameter which, except for the restrictions noted below, may be assigned any function. If the user does not specify a function for a particular key or key sequence, the current value for that key or key sequence does not change. The abbreviated and spelled-out function names are:

*ATTN	Attention
*BCKSPC	Record Backspace
*CLEAR	Clear Screen
*DOWN	Roll Down
*DSPATR	Display Imbedded Attributes
*F1–*F24	F1 through F24 Function Keys
*HELP	5250 Help

*HLP3270	3270 Help Text (Display Active Keyboard Map)
*NONE	No Assignment
*PRINT	Print Screen
*RESET	Error reset
*SAME	No Change
*SYSREQ	System Request
*TEST	Test Request
*UP	Roll up

Additional Considerations

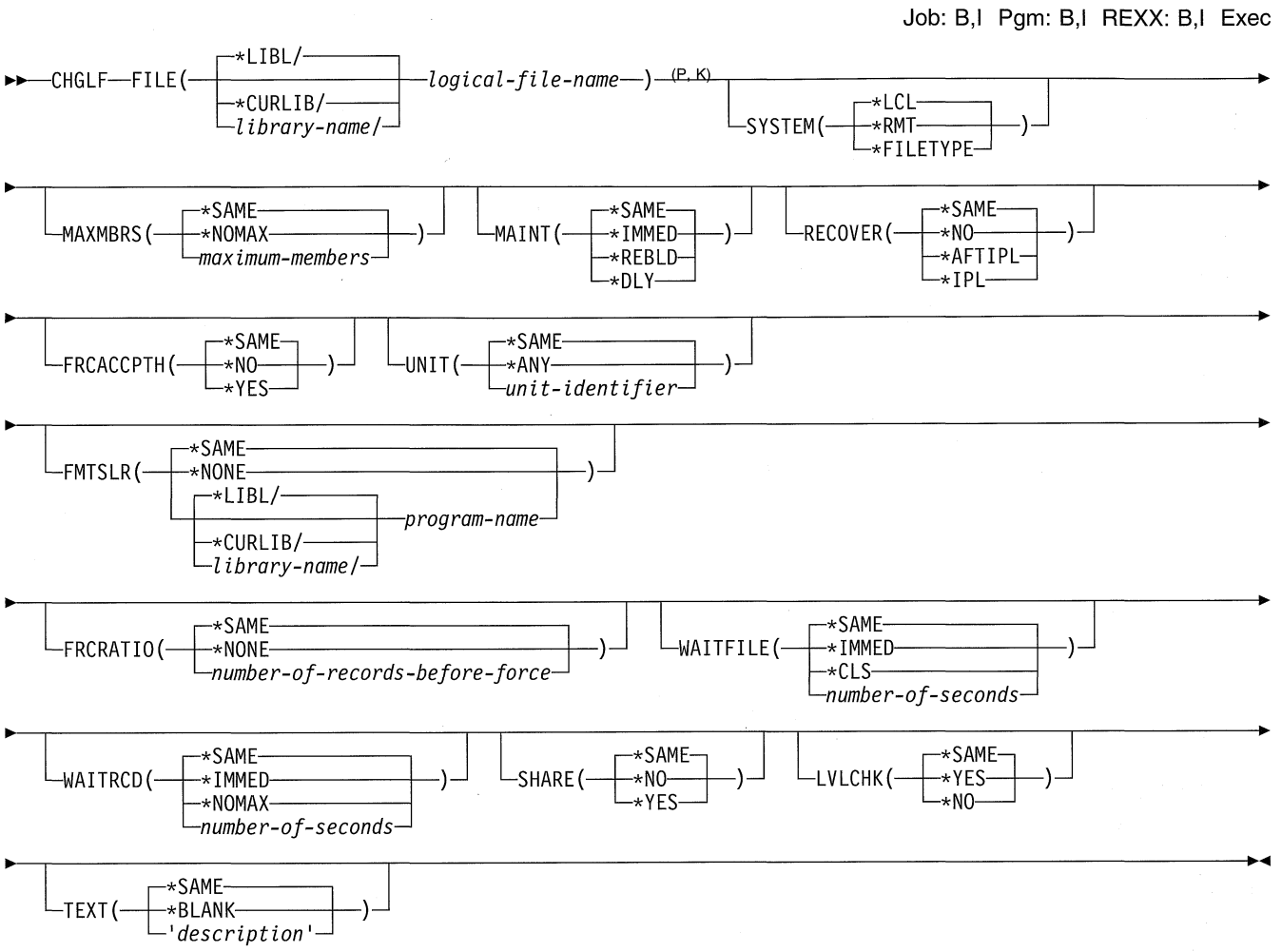
- *HELP, *HLP3270, and *RESET must be assigned to PF1 through PF12, PA1/PF1 through PF12, or PA2/PF1 through PF12. Because these functions are required for 3270 work station device support to function properly and not all keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless that function is also assigned to keys PF1 through PF12, PA1/PF1 through PF12, or PA2/PF1 through PF12.
- It is recommended that *F1 and *SYSREQ be assigned to PF1 through PF12, PA1/PF1 through PF12, or PA2/PF1 through PF12.
- The value *ATTN cannot be explicitly assigned to a 3270 remote attach work station. If the value *ATTN is explicitly chosen, a diagnostic message is sent.

Example

```
CHGKBDMAP   PF1(*F1)  PA1PF1(*HELP)
```

This command would be entered if the user is generally satisfied with the keyboard mapping (either the default mapping or a user-defined mapping) except for one or two PF key assignments. No other PF key sequences are affected by this command. The above command could also be started in the program that started the application (thus tailoring the work station to whatever application is run).

CHGLF (Change Logical File) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Logical File (CHGLF) command changes the attributes of a logical file and its members. The changed attributes are used for all members subsequently added to the file. To change the attributes of a specific member, use the Change Logical File Member (CHGLFM) command.

Restrictions:

1. To change a logical file, the user must have object management authority and object operational authority for the file and read authority to the library.
2. To change the file, an exclusive no read lock is necessary; no one may be using the file for any purpose.

Required Parameter

FILE

Specifies the qualified name of the logical file that is changed.

Note: If a DDM file is specified, then the logical file (specified on the RMTFILE parameter of the Create Distributed Data Management File (CRTDDMF) command) is changed on the remote system specified on the RMTLOCNAME parameter on the CRTDDMF command. More information is outlined in the SYSTEM parameter of this command.

The name of the logical file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

logical-file-name: Specify the name of the logical file that is changed.

Optional Parameters

SYSTEM

Specifies whether the logical file is changed on the local system or on a remote system.

***LCL:** The logical file is changed on the local system.

***RMT:** The logical file is changed on a remote system using distributed data management (DDM). The logical file name specified on the FILE parameter must be the name of the DDM file (created using the Create Distributed Data Management File (CRTDDMF) command). The DDM file contains the name of the logical file to be changed (RMTFILE parameter on the CRTDDMF command) and the name of the remote system (RMTLOCNAME parameter on the CRTDDMF command).

***FILETYPE:** If the name specified on the FILE parameter is a DDM file, the logical file is changed on the remote system specified by the RMTLOCNAME parameter of that DDM file. Otherwise, the name specified on the FILE parameter to be changed must be the name of a local logical file.

MAXMBRS

Specifies the maximum number of members that the logical file can contain at any time. The maximum number of members specified must be greater than or equal to the current number of members in the file.

***SAME:** The value does not change.

***NOMAX:** The system maximum is used.

maximum-members: Specify the maximum number of members that the logical file can contain. Valid values range from 1 through 32767.

MAINT

Specifies the type of access path maintenance used for all members of the logical file. This parameter is valid only for keyed files or join files.

Note: For a join logical file, this parameter value applies to all join secondary files, even if the join file is not a keyed file.

***SAME:** The value does not change.

***IMMED:** The access path is maintained for each physical file member whether the source physical file is opened or closed. The access path is changed whenever a record is updated, added to, or deleted from a member of this file or a logical file member based on a member of this file.

***REBLD:** The access path is rebuilt when a file member is opened. The access path is continuously maintained until the member is closed; then access path maintenance is ended. *REBLD is not valid for keyed access path that contain unique key values.

***DLY:** The maintenance of the access path is delayed until the member is opened for use. The access path is then changed only for records that are added, deleted, or changed since the file was last closed. (While the file is open, all changes made to based-on members are immediately reflected in the keyed access path of the members of the opened files, no matter what is specified for the MAINT parameter.) To prevent a lengthy rebuild time when the file is opened, *DLY is specified only when the number of changes to the access path between a close operation and the next open operation are small (when key fields in records for this access path change infrequently). *DLY is not valid for keyed access paths that require unique key values.

If the number of changes saved reaches approximately 10% of the access path size, the system stops saving changes and the access path is completely rebuilt the next time the file is opened.

RECOVER

Specifies, for files having immediate or delayed maintenance on their keyed access path, when recovery processing of the file is done if a system failure occurs while the keyed access path is being changed.

A keyed access path having immediate or delayed maintenance is rebuilt during IPL (before any user can run a job), after the IPL, or when the file is next opened. While the access path is being rebuilt, the file cannot be used by any job.

An access path having rebuild maintenance is rebuilt the next time its file is opened, the time that it is normally rebuilt. This parameter is not valid if the file does not have a keyed access path.

***SAME:** The value does not change.

***NO:** The access path of the file is not rebuilt. The file's access path, if not valid, is rebuilt the next time the file is opened.

***AFTIPL:** The file has its access path rebuilt after the IPL operation is completed. This option allows other jobs not using this file to start processing immediately after the IPL has been completed. If a job tries to open the file while its access path is being rebuilt, a file open exception occurs.

***IPL:** The file has its access path rebuilt during the IPL operation. This ensures that the access path of the file is rebuilt before the first user program tries to use it; however, no jobs start running until after all files that specify RECOVER(*IPL) have their keyed access paths rebuilt.

FRCACCPTH

Specifies, for files with keyed access paths only, whether access path changes are forced to auxiliary storage along with the associated records in the file. FRCACCPTH(*YES) minimizes (but does not remove) the possibility that an abnormal job end can cause damage to the access path, which then requires it to be rebuilt.

Note: For a join logical file, this parameter value applies to all join secondary files even if the join file is not a keyed file.

***SAME:** The value does not change.

***NO:** The access path and changed records are not forced to auxiliary storage whenever the access path is changed.

***YES:** The access path and changed records are forced to auxiliary storage whenever the access path is changed. If this value is specified, MAINT(*REBLD) is not valid.

FRCACCPTH(*YES) slows the response time of the system if the access path is changed in an interactive job. If the access path is changed frequently, the general performance of the system is affected.

UNIT

Specifies the unit identifier of the auxiliary storage unit where the system tries to allocate the storage space for the access path of the file.

Notes:

1. The system attempts to put any new increments for an access path on the specified unit. However, the space for the access path allocated before the change in the unit specification continues to be on the unit where it was originally allocated. To have the whole access path reside on the specified unit, save the member, delete the member from the system, and then restore it to the system.
2. Join logical file applies to all join secondary files, even if the join file is not a keyed file. If a unit value is specified for a join logical file, the storage space for the access paths of all of the join secondary files are allocated on the specified unit, even if the join file is not a keyed file.

***SAME:** The value does not change.

***ANY:** The storage space for the file and its members is allocated on any available auxiliary storage unit.

unit-identifier: Specify the storage unit on which the system tries to allocate the storage space for the access path of the file.

Device Type	Unit	Unit Identifier
62PC	1-6	1-6
3370 9330S	Module 1, actuator 1 Module 1, actuator 2 Module 2, actuator 1 Module 2, actuator 2 Module 3, actuator 1 Module 3, actuator 2 Module 4, actuator 1 Module 4, actuator 2 Module 5, actuator 1 Module 5, actuator 2 Module 6, actuator 1 Module 6, actuator 2 Module 7, actuator 1 Module 7, actuator 2 Module 8, actuator 1 Module 8, actuator 2	7 8 9 10 11 12 13 14 17 18 19 20 21 22 23 24

Note: These identifiers remain the same for systems that have 3370 devices and fewer than six 62PC devices.

FMTSLR

Specifies the qualified name of a record format selector program that is called when the logical file member contains more than one logical record format. The user-written selector program is called when a record is inserted into the database file and a record format name is not included in the high-level-language program. The selector program receives the record as input, determines the record format used, and returns the record format name to the database. This program must do this function for every member in the logical file that has more than one record format, unless the high-level-language program itself specifies the record format name. More information on the use of format selector programs is in the *Database Guide*.

This parameter is not valid if the logical file has only one record format.

***SAME:** The value does not change.

***NONE:** There is no selector program for this logical file. The file cannot have more than one logical record format, or the high-level-language program itself must specify the record format name.

The name of the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the format selector program that is called when a record is inserted into a

member having more than one format. The selector program name can be optionally qualified by the name of the library where the program is stored.

If USRPRF(*OWNER) is specified in the create program command, the program that is created cannot be used as a format selector program.

FRCRATIO

Specifies the number of inserted, changed, or deleted records that are processed before they are forced to auxiliary (permanent) storage. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

The force write ratio specified for a logical file is less than or equal to the smallest force write ratio of its based-on files. If a larger force write ratio is specified, it is ignored and a message is sent informing the user of the action.

If a physical file associated with this logical file is being journaled, a larger force write ratio or *NONE may be specified. More information on the Journal Management Facility is in the *Advanced Backup and Recovery Guide*.

***SAME:** The value does not change.

***NONE:** There is no force write ratio; the system determines when the records are written in auxiliary storage.

number-of-records-before-force: Specify the number of new or changed records that are processed before they are explicitly forced into auxiliary storage.

WAITFILE

Specifies the number of seconds that the program waits for the file resources and session resources to be allocated when the file is opened, or for the device or session resources to be allocated when an acquire operation is performed to the file. If those resources are not allocated within the specified wait time, an error message is sent to the program. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

Note: An immediate allocation of the device by the device resource is required when an acquire operation is performed to the file.

***SAME:** The value does not change.

***IMMED:** The program does not wait; when the file is opened, an immediate allocation of the file resources is required.

***CLS:** The job default wait time is used as the wait time for the file resources being allocated.

number-of-seconds: Specify the number of seconds that the program waits for the file resources to be allocated to the job. Valid values range from 1 through 32767 seconds.

WAITRCD

Specifies the number of seconds that a program waits for a record to be updated or deleted, or for a record read in the commitment control environment with LCKLVL(*ALL) specified. More information on record locking is in the *Database Guide*. If the record is not allocated in the specified wait time, an error message is sent to the program.

***SAME:** The value does not change.

***IMMED:** The program does not wait; when a record is locked, an immediate allocation of the record is required.

***NOMAX:** The system maximum is used.

number-of-seconds: Specify the number of seconds that the program waits for the record to be allocated to the job. Valid values range from 1 through 32767 seconds.

SHARE

Specifies whether the open data path (ODP) for the logical file is shared with other programs in the routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

More information on shared database files is in the *Database Guide*.

***SAME:** The value does not change.

***NO:** The ODP created by the program with this attribute is not shared with other programs in the routing step. Every time a program opens the file with this attribute, a new ODP to the file is created and activated.

***YES:** The ODP created with this attribute is shared with each program in the routing step that also specifies SHARE(*YES) when it opens the file.

Note: When SHARE(*YES) is specified and control is passed to a program, a read operation in that program retrieves the next input record. A write operation produces the next output record.

LVLCHK

Specifies whether the record format level identifiers in the program are checked against those in the logical file when the file is opened. If so, the record format identifiers in the program must match those in the logical file. This value can be overridden by the Override with Database File (OVRDBF) command at run time.

***SAME:** The value does not change.

***YES:** The level identifiers of the record formats are checked when the file is opened. If the level identifiers do not match, an error message is sent to the program requesting the open operation, and the file is not opened.

***NO:** The level identifiers are not checked when the file is opened.

TEXT

Specifies text that briefly describes the logical file. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Examples

Example 1: Changing the Format Selector Program

```
CHGLF  FILE(QGPL/INV)  FMTSLR(INVFMTS)
```

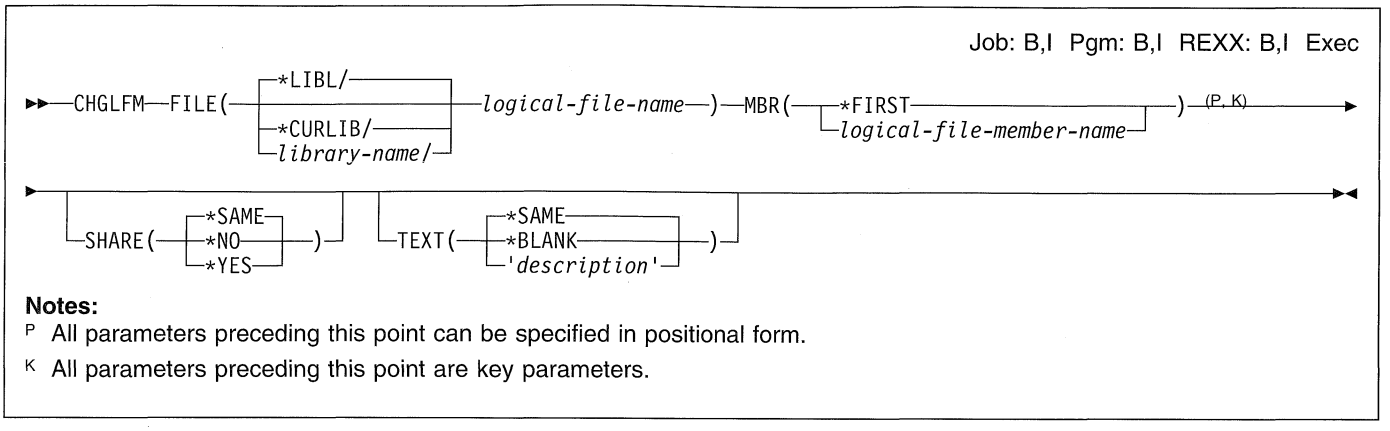
This command changes the logical file INV that is stored in the QGPL library, resulting in the new format selector program, INVFMTS, which is being used with the logical file. *LIBL is used to find the format selector program.

Example 2: Changing a File Located on the Remote System

```
CHGLF  FILE(QGPL/DDMF)  FMTSLR(INVFMTS)
  SYSTEM(*RMT)
```

This command changes the format selector program for file INV located in the QGPL library on the remote system. Prior to specifying this command, this user had created a DDMF file by specifying the command CRTDDMF FILE(QGPL/DDMF) RMTFILE(QGPL/INV) RMTLOCNAME(AS400).

CHGLFM (Change Logical File Member) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Logical File Member (CHGLFM) command changes the attributes of a logical file member.

Restrictions:

1. To change a logical member, object management and object operational authority for the logical file that contains the member, and read authority for the file library, is required.
2. If a user is holding the file for exclusive use, the member cannot be changed. Users working at the same time may have the member open, but changes made to the member are not reflected in any open members. To make the changes effective, the changes in any open members, the user must first close the member (this must be a full close operation if the member is open SHARE(*YES)) and then open it again.

Required Parameter

FILE

Specifies the qualified name of the logical file that contains the member that is changed.

The name of the logical file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

logical-file-name: Specify the name of the logical file.

Optional Parameters

MBR

Specifies the name of the member, or the first member (*FIRST), that is changed.

***FIRST:** The first member in the database file is used.

logical-file-member-name: Specify the name of the logical file member that is changed.

SHARE

Specifies whether the open data path (ODP) for the logical file member is shared with other programs in the routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

More information on shared database files is in the *Database Guide*.

***SAME:** The value does not change.

***NO:** The ODP created by the program with this attribute is not shared with other programs in the routing step. Every time a program opens the file with this attribute, a new ODP to the file is created and activated.

***YES:** The ODP created with this attribute is shared with each program in the routing step that also specifies SHARE(*YES) when it opens the file.

Note: When SHARE(*YES) is specified and control is passed to a program, a read operation in that program retrieves the next input record. A write operation produces the next output record.

TEXT

Specifies text that briefly describes the logical file member. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

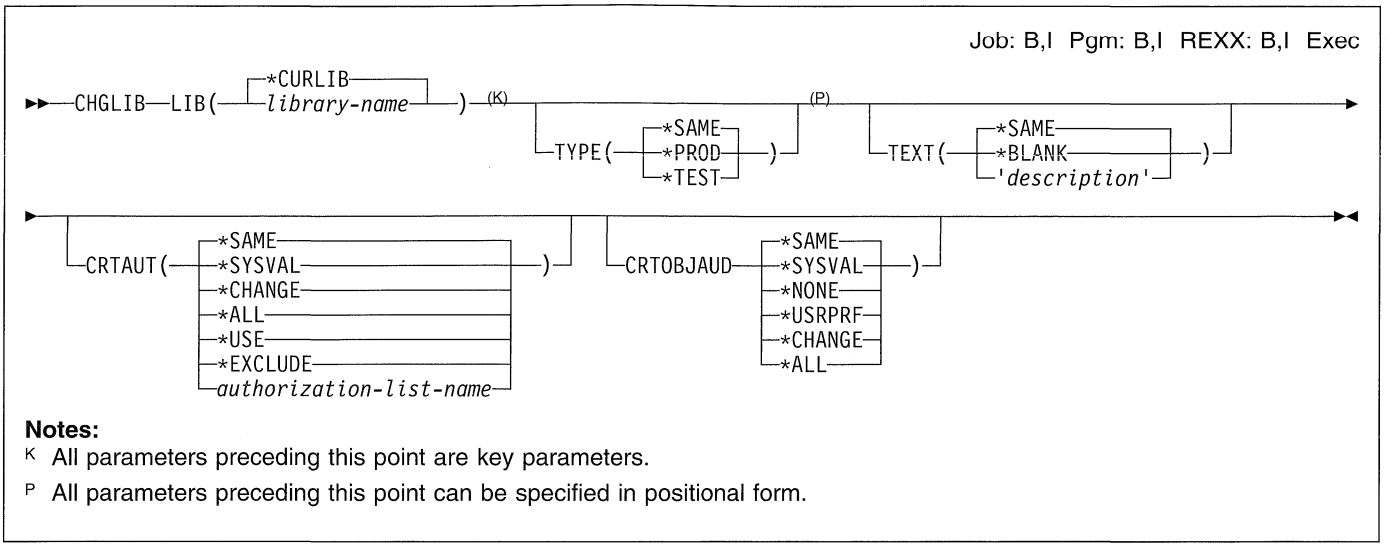
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGLFM   FILE(*CURLIB/INV)  MBR(FEB)
  TEXT('Logical file member for FEB')
```

The command changes the text for the member named FEB in the logical file INV that is found in the specified current library.

CHGLIB (Change Library) Command



Purpose

The Change Library (CHGLIB) command changes the type attribute, the text description of a library, the default create authority value of a library, and the default object auditing value of a library.

Restrictions:

1. You must have *OBJMGT authority to the library that is changed and *READ authority to the library QSYS.
2. You must have *AUDIT special authority to specify a value other than *SAME on the CRTOBJAUD parameter.

Required Parameter

LIB

Specifies the name of the library to be changed.

***CURLIB:** The current library for the job is changed. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be changed.

Optional Parameters

TYPE

Specifies the new library type.

***SAME:** The value does not change.

***PROD:** This is a production library. Database files in production libraries are not opened for updating if a user is in debug mode and he requested that production libraries be protected. A user can protect all database files in production libraries by specifying UPDPROD(*NO) on the Start Debug (STRDBG) command to begin testing. However, this protection does not prevent the program from deleting database files or from changing other objects (such as data areas) in the library.

***TEST:** This is a test library. All objects in a test library are updated during testing, even if special protection is requested for production libraries.

TEXT

Specifies text that briefly describes the library and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

CRTAUT

Specifies, when an object is created into this library, the authority given to users who do not have specific authority to the object (public authority), who are not on the authorization list, and whose user group has no specific authority to the object.

***SAME:** The value does not change.

***SYSVAL:** The QCRTAUT system value is used.

***CHANGE:** The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority.

***ALL:** The user can perform all operations on the object except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user cannot transfer ownership of the

object. If the object is an authorization list, the user cannot add, change, or remove user IDs.

***USE:** The user can perform basic operations on the library, such as running a program or reading a file. The user cannot change the library. *USE authority provides object operational authority and read authority.

***EXCLUDE:** The user cannot access the library.

authorization-list-name: Specify the name of the authorization list whose authority is used for the object.

CRTOBJAUD

Specifies the auditing value for objects created in this library.

***SAME:** The value does not change.

***SYSVAL:** The value specified in the system value QCRTOBJAUD is used.

***NONE:** Using or changing this object will not cause an audit entry to be sent to the security journal.

***USRPRF:** The user profile of the user accessing this object is used to determine if an audit record will be sent for this access. The OBJAUD keyword of the CHGUSRAUD command is used to turn on auditing for a specific user.

***CHANGE:** All change accesses to this object by all users are logged.

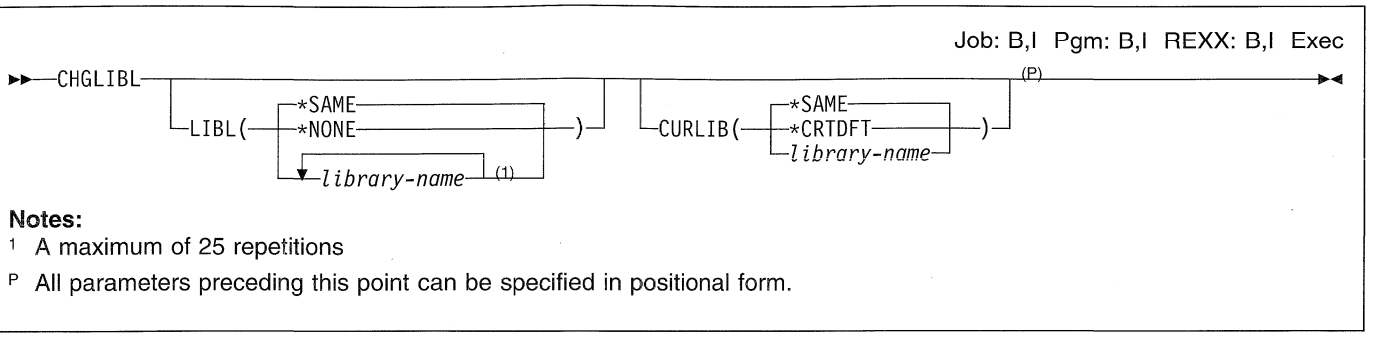
***ALL:** All change or read accesses to this object by all users are logged.

Example

```
CHGLIB   LIB(LIB1) TYPE(*TEST)
  TEXT('NEW TEXT FOR LIB1')
```

This command changes the TYPE attribute of the library named LIB1 to *TEST and changes the text description to the value specified by the TEXT parameter.

CHGLIBL (Change Library List) Command



Purpose

The Change Library List (CHGLIBL) command changes the user's portion of the current job's library list with the list of libraries specified by the user. This command does not affect the system portion of the library list, nor does it affect any other job's library list.

Restrictions: The user must have *USE authority for all the specified libraries before the job's current library list is changed. If the user is not authorized to one of the libraries, the command is not run.

Optional Parameters

LIBL

Specifies the libraries that are placed in the user's portion of the current job's library list.

***SAME:** The value does not change.

***NONE:** No libraries are in the user's portion of the job's library list.

library-name: Specify the names of the libraries in the order in which they are searched.

CURLIB

Specifies the library that replaces the current library entry in the job's library list.

***SAME:** The value does not change.

***CRTDFT:** No library is in the current library entry of the library list. If objects are created into the current library, the QGPL library is used as the default current library.

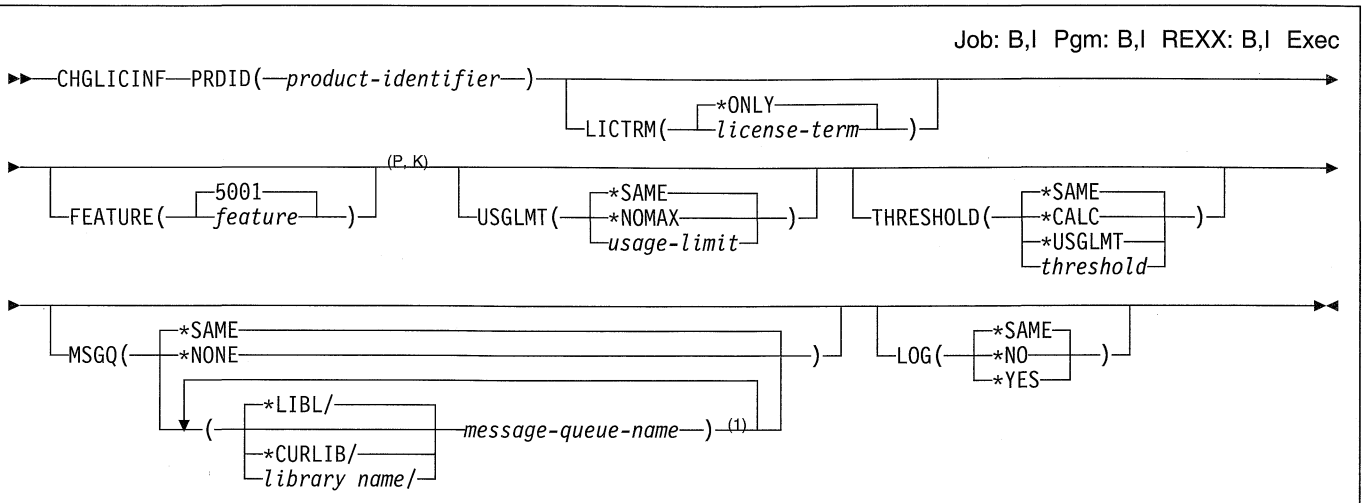
library-name: Specify the name of the library that replaces the current library entry in the job's library list.

Example

```
CHGLIBL   LIBL(ULIB10  ULIB15  QGPL)
```

This command changes the user's portion of the current job's library list that existed before this command was entered. The new library list contains libraries ULIB10, ULIB15, and QGPL. They are searched in that order.

CHGLICINF (Change License Information) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

1 A maximum of 5 repetitions.

Purpose

The Change License Information (CHGLICINF) command allows a user to change license information for a product or feature that contains license information. You can use this command to change the following:

- The usage limit
- The threshold
- The users to notify when the threshold is exceeded, when an attempt to exceed the usage limit is made, or when the usage limit has been changed
- The logging of requests to use the product or feature when the usage count is equal to the usage limit

If the usage limit is increased, an order form is spooled. The order form can be mailed to the software provider to begin the ordering and billing processes.

Restrictions: This command is shipped with public *EXCLUDE authority.

Required Parameter

PRDID

Specifies the seven-character identifier (ID) of the product for which license information is to be changed.

Optional Parameters

LICTRM

Specifies the license term for which license information is to be changed.

***ONLY:** Only one license term of the product is on the system and that license term identifies the product to be changed.

license-term: Specify the license term of the product to be changed in one of the following formats:

Vx
VxRx
VxRxMy

where values for *x* range from 0 through 9, and values for *y* range from 0 through 9 and from A through Z. Use the Work with License Information (WRKLICINF) command to determine the license term.

FEATURE

Specifies the feature of the product specified on the PRDID parameter for which license information is to be changed. You can use the Work with License Information (WRKLICINF) command to identify the feature number of the product.

5001: The license information for the feature 5001 is to be changed.

feature: Specify the feature for which license information is to be changed. Valid values range from 5001 through 9999.

USGLMT

Specifies the maximum number of users (usage limit) for this product or feature. The software provider sets the original value on this parameter. For concurrent usage, this is the maximum number of jobs allowed to access the product or feature at any given time. For registered

usage, this is the maximum number of license users that can be registered to use this product or feature.

***SAME:** The value does not change.

***NOMAX:** The number of users is not limited.

usage-limit: Specify the maximum number of users for this product or feature. Valid values range from 0 through 999999.

THRESHOLD
Specifies the threshold for this product or feature. When the number of users for this product exceeds the threshold, a message is sent to the system operator's message queue (QSYS/QSYSOPR) and to the message queues specified on the MSGQ parameter.

Note: If you specify a value equal to the usage limit, no threshold messages are sent. Messages are sent only when an attempt to exceed the usage limit is made, or when the usage limit is changed.

***SAME:** The value does not change.

***CALC:** The system calculates the threshold for this product or feature. The calculated threshold is equal to 90 percent of the usage limit (maximum number of users specified on the USGLMT parameter). If the usage limit is 1 or *NOMAX, the threshold is equal to the usage limit.

***USGLMT:** The threshold for this product or feature is equal to its usage limit.

threshold: Specify the threshold for this product or feature. Valid values range from 0 through 999999.

MSGQ
Specifies the qualified names of a maximum of five message queues to which messages are sent if (1) the threshold is exceeded, (2) an attempt to exceed the usage limit is made, or (3) the usage limit is changed. These messages are also sent to the system operator's message queue (QSYS/QSYSOPR).

***SAME:** The value does not change.

***NONE:** Messages are sent only to the system operator's message queue (QSYS/QSYSOPR).

The name of the message queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

message-queue-name: Specify the name of the message queue to which messages are sent.

LOG
Specifies whether to record requests to use the product or feature when the usage count is equal to the usage limit. If requests are recorded, they are recorded in the QLZALOG journal in QUSRSYS.

***SAME:** The value does not change.

***NO:** The requests made when the usage count is equal to the usage limit are not recorded.

***YES:** The requests made when the usage count is equal to the usage limit are recorded. When a license request is made and the usage count is equal to the usage limit, the user requesting a license and all current license users are recorded in the QLZALOG journal in QUSRSYS.

Examples

Example 1: Changing the Usage Limit for a Product with One License Term on the System

```
CHGLICINF  PRDID(1MYPROD) LICTRM(*ONLY) USGLMT(35)
  THRESHOLD(30) MSGQ(MYLIB/MYMSGQ) LOG(*NO)
```

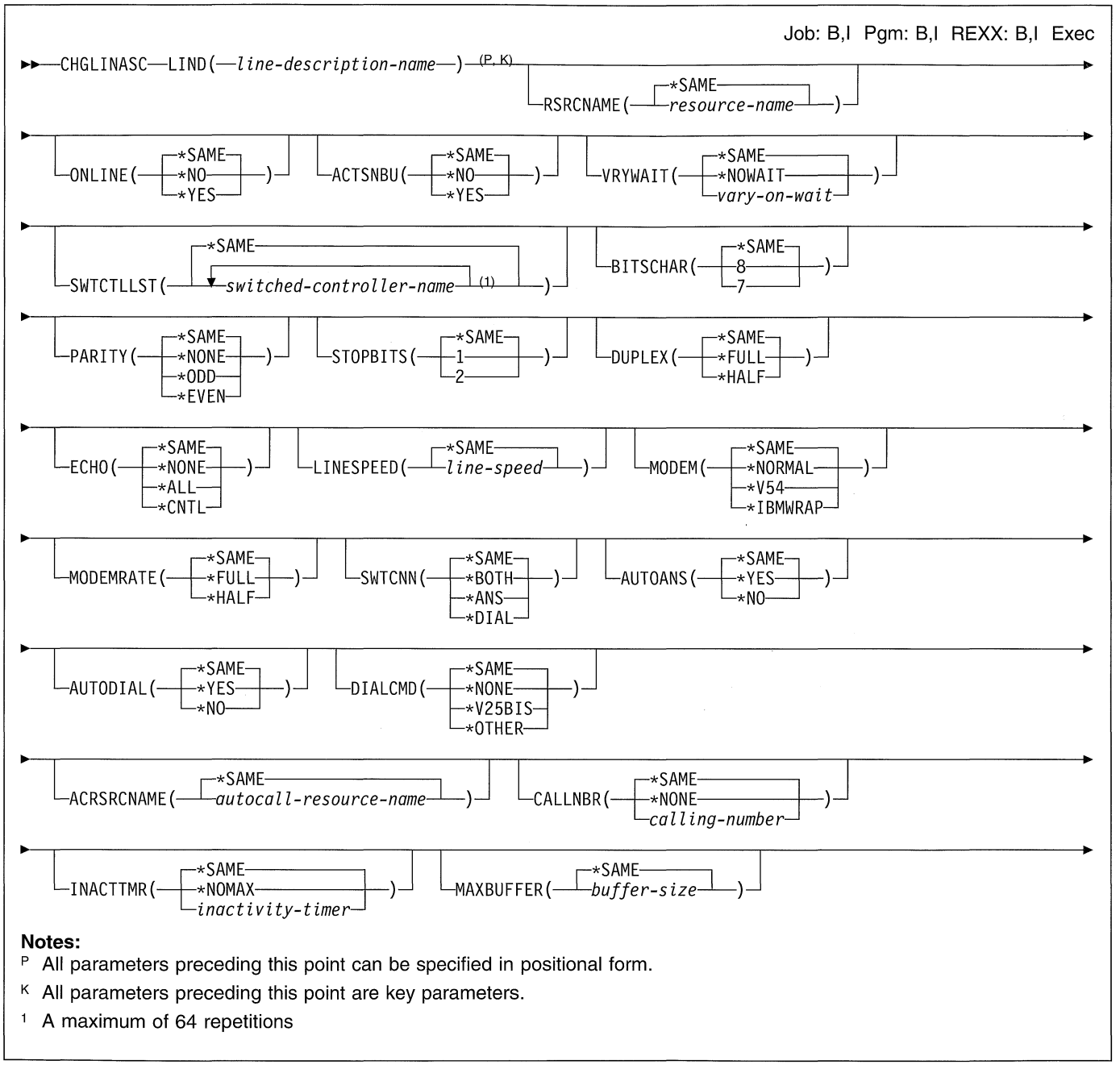
This command allows you to change the usage limit for the product with product identifier 1MYPROD which has only one license term installed on this system. After this command is run, 35 users will be authorized to use this product. When the usage count reaches 31, messages will be sent to the system operator's message queue and the message queue MYMSGQ in the library MYLIB stating that the threshold has been exceeded. Requests for a license when the usage count is equal to the usage limit will not be recorded. However, changes to the usage limit will be recorded in the QLZALOG journal in QUSRSYS.

Example 2: Changing the Usage Limit for a Product with More Than One License Term on the System

```
CHGLICINF  PRDID(2MYPROD) LICTRM(V2) USGLMT(50)
  THRESHOLD(*USGLMT) MSGQ(*NONE) LOG(*YES)
```

This command allows you to change the usage limit for the product with product identifier 2MYPROD and license term Version 2. After this command is run, 50 users will be authorized to use this product. No threshold messages will be sent. (However, messages will be sent to the system operator whenever the usage limit is changed or the usage count reaches or exceeds the usage limit.) A request for a license when the usage count is equal to the usage limit will be recorded in the QLZALOG journal in QUSRSYS.

CHGLINASC (Change Line Description (Async)) Command

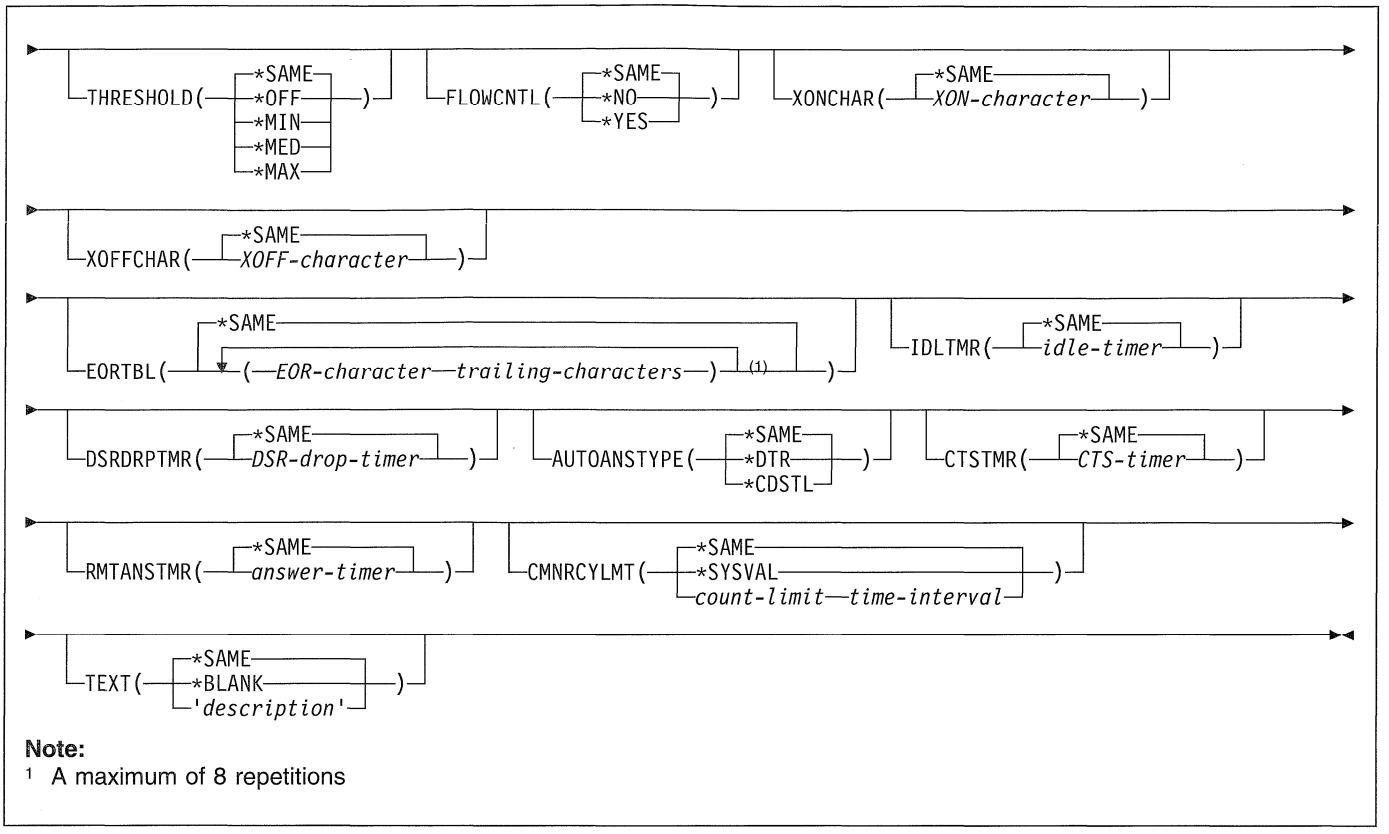


Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

1 A maximum of 64 repetitions



Note:
1 A maximum of 8 repetitions

Purpose

The Change Line Description (Async) (CHGLINASC) command changes a line description for an asynchronous line.

Required Parameter

LIND
Specifies the name of the line description being changed.

Optional Parameters

RSRCNAME
Specifies the resource name that describes the automatic call unit port.

Note: Use the Work with Hardware Resources (WRKHDWRSC) command with *CMN specified for the TYPE parameter to help determine the resource name.

***SAME:** The value does not change.

resource-name: Specify the resource name of the communications port. The resource name consists of the input/output adapter (IOA) resource name and the port number on the IOA. For example, if the resource name of the IOA is LIN01 and the port on the IOA is 2, then the resource name would be LIN012.

ONLINE
Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***NO:** This line is not automatically varied on at IPL.

***YES:** The line is automatically varied on at IPL.

ACTSNBU
Specifies, for controllers supporting the switched network backup (SNBU) feature, whether the SNBU feature is activated or deactivated. Both the local and remote modems must support the SNBU feature to perform a valid activation.

***SAME:** The value does not change.

***NO:** The switched network backup (SNBU) feature is not activated.

***YES:** The switched network backup feature (SNBU) is activated. To start the feature when the nonswitched connection is broken, specify ACTSNBU(*YES) on the line description command.

VRYWAIT
Specifies whether the line is varied on asynchronously or synchronously. For synchronous vary on, specifies how long the system waits for the vary on to complete.

***SAME:** The value does not change.

***NOWAIT:** The system does not wait for the vary on to complete. The line is varied on asynchronously.

vary-on-wait: Specify the time (in seconds) to wait. Valid values range from 15 through 180. The system waits until the line is varied on, or until the specified time passes, before completing the Vary Configuration (VRYCFG) command.

Notes:

1. Specifying a wait time in the line description affects system IPL time, if ONLINE(*YES) is used, by the amount of time it takes to synchronously vary on the line or reach the wait-time value.
2. The time required to vary on a line is the time it takes to put tasks in place to manage the line, to activate the communications I/O processor (IOP) (including downloading the IOP model-unique Licensed Internal Code), and to establish communications with the data circuit-terminating equipment (DCE). Normal vary-on time ranges from 5 through 45 seconds, but can be longer, depending on the system, line protocol, and other factors.

SWTCTLLST

Specifies, for switched lines, or those with the SNBU feature, the names of up to 64 switched controllers that can establish a connection with this switched asynchronous line. The controller descriptions must already exist.

***SAME:** The value does not change.

switched-controller-name: Specify the switched controller name. The user can specify up to 64 switched controllers that can establish a connection.

BITSCHAR

Specifies the number of data bits per character (excluding the parity bit, if any).

***SAME:** The value does not change.

8: A value of eight data bits per character is used.

7: Seven data bits per character are used.

PARITY

Specifies the type of parity used for error checking. A parity bit is a binary digit inserted in each byte of data to make the arithmetic sum of all the digits, including the parity bit, always odd or always even.

Note: The remote system must use the parity type specified by the PARITY parameter.

***SAME:** The value does not change.

***NONE:** No parity bit is inserted in the data byte.

***ODD:** The arithmetic sum of all the digits, including the parity bit, is odd.

***EVEN:** The arithmetic sum of all the digits, including the parity bit, is even.

STOPBITS

Specifies the number of stop bits added to the end of each character. These bits are used to keep the local and remote ends of the line synchronized. The local system and the remote system must use the same number of stop bits.

Note: At line speeds of 300 bps or less, two stop bits are recommended.

***SAME:** The value does not change.

1: Add one stop bit to each character.

2: Two stop bits are used.

DUPLEX

Specifies whether request-to-send (RTS) is permanently turned on (for full-duplex modems) or turned on only when data transmission is required (for half-duplex modems).

***SAME:** The value does not change.

***FULL:** Request-to-send (RTS) is permanently turned on (for full-duplex modems).

***HALF:** RTS is turned on only when transmission is required (for half-duplex modems).

ECHO

Specifies whether the system echoes (sends back) to the remote system all characters it receives, all characters except end-of-record characters, or whether echo is inhibited.

Specify *ALL or *CNTL only when communicating with a remote system that requires echo. If *ALL or *CNTL is specified for the echo prompt, *FULL must also be specified for the duplex prompt.

***SAME:** The value does not change.

***NONE:** No characters received are echoed to the remote system.

***ALL:** All characters received are echoed to the remote system.

***CNTL:** All characters received before end-of-record characters are echoed to the remote system.

LINESPEED

Specifies the line speed in bits per second (bps).

***SAME:** The value does not change.

line-speed: Specify the line speed. Valid values are: 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 7200, 9600, or 19200 bps.

MODEM

Specifies the type of modem supported on the communications line. Refer to the modem documentation to determine the appropriate value.

***SAME:** The value does not change.

***NORMAL:** No attempt is made to run diagnostic tests on the modem.

***V54:** This value means that certain types of diagnostics (as defined by the CCITT recommendations) are run to the modem. The AS/400 system supports CCITT

V.54 loop 3 (local loop back) and loop 2 (remote loop back).

***IBMWRAP:** An IBM modem with wrap test capabilities is used on the communications line.

MODEMRATE
Specifies the speed at which the line operates if the line has the data rate select function.

Note: The user is responsible for ensuring that the specified line speed corresponds to the actual modem rate.

***SAME:** The value does not change.

***FULL:** The line operates at the full rate of the modem.

***HALF:** The line operates at half the rate, or at the alternate rate of the modem.

SWTCNN
Specifies whether the switched or switched network backup line is used for incoming calls, for outgoing calls, or for both.

***SAME:** The value does not change.

***BOTH:** The line is used for both incoming and outgoing calls.

***ANS:** The line is used for incoming calls only.

***DIAL:** The line is used for outgoing calls only.

AUTOANS
Specifies, for switched or switched network backup (SNBU) lines, whether the system automatically answers a call from a remote system to establish the connection or whether the user must manually answer the call and place the modem in data mode.

***SAME:** The value does not change.

***YES:** The incoming call is automatically answered by the automatic answer feature.

Note: *YES is valid only if the modem has the automatic answer feature.

***NO:** The incoming call must be manually answered.

AUTODIAL
Specifies, for switched or switched network backup (SNBU) lines, whether the system automatically calls a remote system to establish a connection or whether the system operator must manually place the call.

Note: *YES is valid only if the system is using an automatic call unit or if the modem being used is capable of calling through a command interface.

***SAME:** The value does not change.

***YES:** The line connection is made by the system automatically dialing the remote system.

***NO:** The line connection is made by manually dialing the remote system.

DIALCMD
Specifies the type of dial command used to establish a switched connection with a remote system.

***SAME:** The value does not change.

***NONE:** No dial command type is specified. An automatic call unit establishes the connection.

***V25BIS:** The dial command type V.25 bis is used. V.25 bis allows the use of one physical interface for call establishment and data transmission. It is referred to as a serial automatic call interface because the digits are presented serially on the link from the system (DTE) to the modem (DCE).

***OTHER:** Another dial command type is used. The IBM AT command set is one example of another dial command type that is used by asynchronous protocols.

ACRSRCNAME
Specifies the resource name that describes the automatic call unit port.

***SAME:** The value does not change.

autocall-resource-name: Specify the resource name of the autocall unit port. The resource name consists of the input/output adapter (IOA) resource name and the port number on the IOA. For example, if the resource name of the IOA is LIN02 and the port is 1, then the resource name is LIN021.

CALLNBR
Specifies the local telephone number of the line used for the V.25 bis call request with identification (CRI) dial command. This parameter is used when the CRI function is needed for V.25 bis. When V.25 bis CRI dialing is used, the system takes the called (connection) number from the CNNNBR parameter of the controller description, adds a separator character (;), and concatenates the calling number at the end. Specify the calling number only if the modem and the network both support the CRI dial command.

***SAME:** The value does not change.

***NONE:** The CRN (Call Request Normal) dial command is used by the V.25 bis line.

calling-number: Specify up to 32 characters that represent the local telephone number for V.25 bis CRI autodialing.

INACTTMR
Specifies the time (in tenths of a second) that the system waits for activity on a switched line before disconnecting.

***SAME:** The value does not change.

***NOMAX:** There is no disconnect limit.

inactivity-timer: Specify an inactivity timer value.

MAXBUFFER
Specifies the maximum size of inbound and outbound data buffers.

***SAME:** The value does not change.

buffer-size: Specify a value ranging from 128 through 4096 characters for the buffer size.

THRESHOLD

Specifies the temporary error threshold level being monitored by the system. A permanent error is reported only if the errors occurred consecutively and exceeded the retry limit.

Note: Specifying the THRESHOLD parameter affects all threshold errors. They cannot be specified individually.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** The error threshold is set at a minimum monitoring level.

***MED:** Error thresholding is set to a medium monitoring level.

***MAX:** The error threshold is set at a maximum monitoring level.

FLOWCNTL

Specifies whether the hardware automatically generates or responds to XON/XOFF characters.

***SAME:** The value does not change.

***NO:** Prevents the hardware from generating or recognizing flow control characters.

***YES:** The hardware recognizes flow control characters. This means that when an XOFF character is received, the hardware stops transmission until an XON character is received. It also means that the hardware sends an XOFF character to the remote location when it is incapable of receiving more than 4 characters. When the hardware is again able to receive at least five characters, it sends an XON character to the remote system.

XONCHAR

Specifies the hexadecimal value of the flow control character XON. If the system receives an XOFF character while sending data, it automatically stops sending, and starts sending data again only after receiving an XON character.

***SAME:** The value does not change.

XON-character: Specify a value ranging from 01 through FF for the hexadecimal value of the flow control character XON. The character should not appear in the normal data stream, such as hexadecimal 20 (ASCII blank).

XOFFCHAR

Specifies the hexadecimal value of the flow control character XOFF. If the system receives an XOFF character while sending data, it automatically stops sending, and starts sending data again only after receiving an XON character.

***SAME:** The value does not change.

XOFF-character: Specify a value ranging from 01 through FF for the hexadecimal value of the flow control character XOFF. The character should not appear in the normal data stream, such as hexadecimal 20 (ASCII blank).

EORTBL

Specifies the end of record (EOR) table, which allows the system to recognize logical records when receiving data. The line feed (LF) can be defined as an EOR character in the data stream and have the hardware return the data when the LF character is detected in the data stream.

The EOR table is specified as a set of pairs, where the first element of a pair is the EOR character and the second element specifies the number of characters that follow the EOR character.

***SAME:** The value does not change.

Element 1: End of Record Characters

EOR-character: Specify a value ranging from hexadecimal 01 through 7F (if 7 bits per character) or 01 through FF (if 8 bits per character). End-of-record characters are specified as they appear on the line after any translation by the asynchronous communications support. Up to eight EOR character entries can be specified.

Element 2: Additional End of Record Characters

trailing-characters: Specify the number of additional characters received after the end-of-record character is detected. The number of trailing characters ranging from 0 through 4.

IDLTMR

Specifies the time (in 0.1 second intervals) that the system waits between characters before the adapter forwards the receive buffer to the system.

***SAME:** The value does not change.

idle-timer: Specify a value ranging from 1 through 254 in 0.5 second intervals, or specify 0 to indicate no timer.

DSRDRPTMR

Specifies the amount of time the system waits for the modem to exit the Data Set Ready (DSR) state before signaling an error.

***SAME:** The value does not change.

DSR-drop-timer: Specify a value ranging from 3 through 60 seconds.

AUTOANSTYP

Specifies the method the system uses to answer incoming calls.

***SAME:** The value does not change.

***DTR:** The system enters the Data Terminal Ready state, signals the modem to answer calls, and waits for the modem to enter the Data Set Ready (DSR) state.

***CDSTL:** The system enters the Connect Data Set to Line (CDSTL) state after monitoring the Ring Indicator to signal the modem to answer the call.

CTSTMR
Specifies the amount of time the system waits for the modem to enter or exit the Clear to Send (CTS) state before signaling an error.

***SAME:** The value does not change.

CTS-timer: Specify a value ranging from 10 through 60 seconds.

RMTANSTMR
Specifies the amount of time the system waits for the modem to enter the Data Set Ready (DSR) state after dialing before signaling an error.

***SAME:** The value does not change.

answer-timer: Specify a value ranging from 30 through 120 seconds.

CMNRCYLMT
Specifies the number of recovery attempts made by the system before an inquiry message is sent to the system operator. Also specifies the time (in minutes) that must elapse before the system sends an inquiry message to the system operator indicating that the recovery attempt count limit is reached.

***SAME:** The value does not change.

***SYSVAL:** The recovery limits specified in the QCMNRCYLMT system value are used.

Element 1: Maximum Recovery Limit

count-limit: Specify the number of second-level recovery attempts to be made. Valid values range from 0 through 99.

Element 2: Recovery Time Interval

time-interval: Specify the time interval (in minutes) at which the specified number of second-level recoveries are attempted. Valid values range from 0 through 120. If the value specified for *count-limit* is not 0, the value 0 specifies infinite recovery.

TEXT
Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

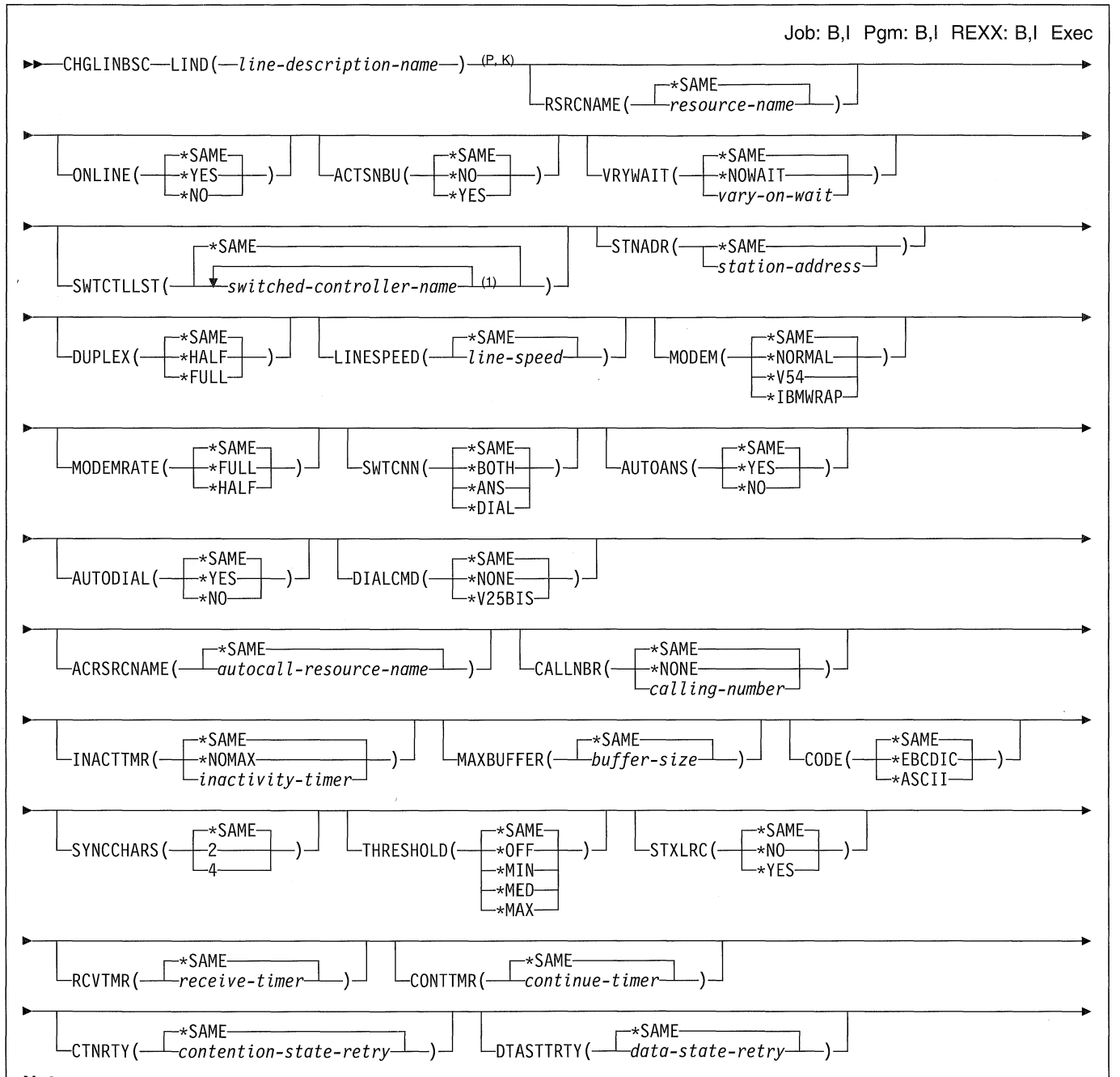
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGLINASC   LIND(CHICAGO)  LINESPEED(4800)
```

This command changes the line speed for line description CHICAGO to 4800 bits per second.

CHGLINBSC (Change Line Description (BSC)) Command

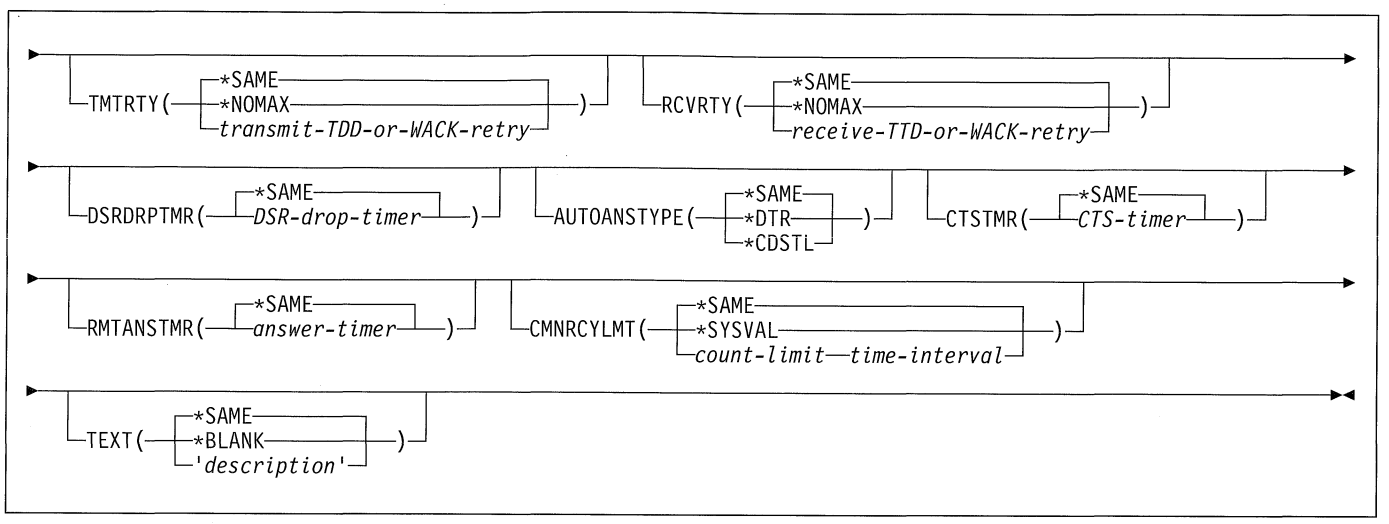


Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

1 A maximum of 64 repetitions.



Purpose

The Change Line Description (BSC) (CHGLINBSC) command changes a line description for a binary synchronous communications (BSC) line.

Required Parameter

LIND

Specifies the name of the line description being changed.

Optional Parameters

RSRCNAME

Specifies the resource name that describes the automatic call unit port.

Note: Use the Work with Hardware Resources (WRKHDWRSC) command with *CMN specified for the TYPE parameter to help determine the resource name.

***SAME:** The value does not change.

resource-name: Specify the resource name of the communications port. The resource name consists of the input/output adapter (IOA) resource name and the port number on the IOA. For example, if the resource name of the IOA is LIN01 and the port on the IOA is 2, then the resource name would be LIN012.

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The line is automatically varied on at IPL.

***NO:** This line is not automatically varied on at IPL.

ACTSNBU

Specifies, for controllers supporting the switched network backup (SNBU) feature, whether the SNBU feature is activated or deactivated. Both the local and remote modems must support the SNBU feature to perform a valid activation.

***SAME:** The value does not change.

***NO:** The switched network backup (SNBU) feature is not activated.

***YES:** The switched network backup feature (SNBU) is activated. To start the feature when the nonswitched connection is broken, specify ACTSNBU(*YES) on the line description command.

VRYWAIT

Specifies whether the line is varied on asynchronously or synchronously. For synchronous vary on, specifies how long the system waits for the vary on to complete.

***SAME:** The value does not change.

***NOWAIT:** The system does not wait for the vary on to complete. The line is varied on asynchronously.

vary-on-wait: Specify the time (in seconds) to wait. Valid values range from 15 through 180. The system waits until the line is varied on, or until the specified time passes, before completing the Vary Configuration (VRYCFG) command.

Notes:

1. Specifying a wait time in the line description affects system IPL time, if ONLINE(*YES) is used, by the amount of time it takes to synchronously vary on the line or reach the wait-time value.
2. The time required to vary on a line is the time it takes to put tasks in place to manage the line, to activate the communications I/O processor (IOP) (including downloading the IOP model-unique Licensed Internal Code), and to establish communications with the data circuit-terminating equipment (DCE). Normal vary-on time ranges from 5 through 45 seconds, but can be longer, depending on the system, line protocol, and other factors.

SWTCTLLST
Specifies, for switched lines, or those with the SNBU feature, this specifies the names of up to 64 switched controllers that can establish a connection with this switched binary synchronous line. The controller descriptions must already exist.

***SAME:** The value does not change.

switched-controller-name: Specify the switched controller name. The maximum number of controllers that can establish a connection with the switched BSC line is 64.

STNADR
Specifies, for multipoint tributary lines, the EBCDIC hexadecimal polling address by which the local system is known to the remote system. The hexadecimal address is the polling address assigned to this AS/400 system.

***SAME:** The value does not change.

station-address: Specify the station address as a 2-digit hexadecimal value ranging from 04 through FE. BSC control characters cannot be specified.

DUPLEX
Specifies whether request-to-send (RTS) is permanently turned on (for full-duplex modems) or turned on only when data transmission is required (for half-duplex modems).

***SAME:** The value does not change.

***HALF:** RTS is turned on only when transmission is required (for half-duplex modems).

***FULL:** Request-to-send (RTS) is permanently turned on (for full-duplex modems).

LINESPEED
Specifies the line speed in bits per second (bps).

***SAME:** The value does not change.

line-speed: Specify the line speed. Valid values are: 600, 1200, 2400, 4800, 7200, 9600, 14400, 19200, 48000, 56000, or 57600 bits per second.

MODEM
Specifies the type of modem supported on the communications line. Refer to the user's modem documentation to determine the appropriate value.

***SAME:** The value does not change.

***NORMAL:** No attempt is made to run diagnostic tests on the modem.

***V54:** A certain type of diagnostics (as defined by the CCITT recommendations) is run to the user's modem. The AS/400 system supports CCITT V.54 loop 3, local loop back, and loop 2, which is a remote loop back.

***IBMWRAP:** An IBM modem with wrap test capabilities is used on the communications line.

MODEMRATE
Specifies the speed at which the line operates if the line has the data rate select feature.

***SAME:** The value does not change.

***FULL:** The line operates at the full rate of the modem.

***HALF:** The line operates at half the full rate, or at the alternate rate of the modem.

SWTCNN
Specifies whether the switched or switched network backup line is used for incoming calls, outgoing calls, or both.

***SAME:** The value does not change.

***BOTH:** The line is used for both incoming and outgoing calls.

***ANS:** The line is used for incoming calls only.

***DIAL:** The line is used for outgoing calls only.

AUTOANS
Specifies, for switched or switched network backup (SNBU) lines, whether the system automatically answers a call from a remote system to establish the connection or whether the user must manually answer the call and place the modem in data mode.

***SAME:** The value does not change.

***YES:** The incoming call is automatically answered by the automatic answer feature.

***NO:** The incoming call must be manually answered.

Note: *YES is valid only if the modem has the automatic answer feature.

AUTODIAL
Specifies, for switched or switched network backup (SNBU) lines, whether the system automatically calls a remote system to establish a connection or whether the system operator must manually place the call.

Note: *YES is only valid if the system is using an automatic call unit and *YES was specified on the AUTOCALL parameter for the CRTLINBSC command.

***SAME:** The value does not change.

***YES:** The system automatically calls a remote system.

***NO:** The system operator must manually call a remote system.

DIALCMD
Specifies the type of dial command used to establish a switched connection with a remote system.

***SAME:** The value does not change.

***NONE:** No dial command is used. An autocall unit is used to establish the connection.

***V25BIS:** V.25 bis is a recommended dial command which allows the use of one physical interface for call establishment and data transmission. It is referred to as

a serial automatic call interface because the digits are presented serially on the link from the systems to the modem.

ACRSRCNAME
Specifies the resource name that describes the automatic call unit port.

***SAME:** The value does not change.

autocall-resource-name: Specify the resource name of the autocall unit port. The resource name consists of the input/output adapter (IOA) resource name and the port number on the IOA. For example, if the resource name of the IOA is LIN02 and the port is 1, then the resource name is LIN021.

CALLNBR
Specifies the local telephone number of the line used for the V.25 bis call request with identification (CRI) dial command. This parameter is used when the CRI function is needed for V.25 bis. When V.25 bis CRI dialing is used, the system takes the called (connection) number from the CNNNBR parameter of the controller description, adds a separator character (;), and concatenates the calling number at the end. Specify the calling number only if the modem and the network both support the CRI dial command.

***SAME:** The value does not change.

***NONE:** CRN dialing is used.

calling-number: Specify the local phone number if V.25 bis CRI dialing is required. Up to 32 characters can be specified. More information on the values allowed by the modem is in the modem documentation.

For short hold mode:

- If the short hold node type is *T20, up to 18 characters can be specified.
- If the short hold node type is *T21, up to 14 characters can be specified.

INACTTMR
Specifies the time (in tenths of a second) that the system waits for activity on a switched line before disconnecting.

***SAME:** The value does not change.

***NOMAX:** There is no disconnect limit.

inactivity-timer: Valid values range from 150 through 4200 in 0.1 second intervals.

MAXBUFFER
Specifies the maximum inbound and outbound data buffers in bytes.

***SAME:** The value does not change.

buffer-size: Specify a buffer size ranging from 8 through 8192 characters.

CODE
Specifies the character code used. The code can be either extended binary-coded decimal interchange code (*EBCDIC) or the American National Standard Code for Information Interchange (*ASCII).

***SAME:** The value does not change.

***EBCDIC:** The extended binary-coded decimal interchange code (EBCDIC) character set code is used.

***ASCII:** The ASCII character set code is used.

SYNCCHARS
Specifies the number of BSC SYN (synchronous) control characters to send when transmitting. The SYN control character is used to establish and maintain synchronization and it is also used as a time fill in the absence of any data or other control character.

***SAME:** The value does not change.

2: The synchronization pattern consists of two consecutive SYN characters. The 9404 System Unit always uses two characters.

4: The synchronization pattern consists of four consecutive SYN characters.

THRESHOLD
Specifies the temporary error threshold level being monitored by the system. A permanent error is reported only if the errors occurred consecutively and exceeded the retry limit.

Note: Specifying the THRESHOLD parameter affects all threshold errors. They cannot be specified individually.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** The error threshold is set at a minimum monitoring level.

***MED:** Error thresholding is set to a medium monitoring level.

***MAX:** The error threshold is set at a maximum monitoring level.

STXLRC
Specifies whether the start of text (STX) control character is included in the longitudinal redundancy check (LRC) calculation. This applies only to lines using the ASCII character code.

***SAME:** The value does not change.

***NO:** The STX control character is not included in the LRC calculation.

***YES:** The STX control character is included in the LRC calculation. The 9404 System Unit always includes STX.

RCVTMR
Specifies the time the system waits for data from the remote system before a receive timeout occurs.

***SAME:** The value does not change.

receive-timer: Specify a value ranging from 30 through 254 in 0.1 second intervals.

CONTTMR

When the system is not ready to transmit or receive data, this parameter specifies the amount of time the system waits before sending a control character that prevents the line from being made inoperative.

***SAME:** The value does not change.

continue-timer: Specify a value ranging from 16 through 24 in 0.1 second intervals.

CTNRTY

Specifies the number of contention state retries to attempt before indicating the error and making the line inoperative.

For BSC, contention is the state that exists after the end-of-transmission (EOT) character is received or sent and before a starting sequence (ENQ) has been positively acknowledged (ACK0).

***SAME:** The value does not change.

contention-state-retry: Specify a value ranging from 0 through 21 for the number of contention state retries.

DTASTTRTY

Specifies the number of data state retries to attempt before indicating the error and making the line inoperative.

For BSC, data state is the time during which BSC is sending or receiving text on the communications line.

***SAME:** The value does not change.

data-state-retry: Specify a value ranging from 0 through 255 for the number of data state retries.

TMTRTY

Specifies the number of retries for transmitting temporary text delay (TTD) or wait before transmitting (WACK) to hold up the line before indicating a session failure. This parameter cannot be specified for an application type of RJE.

***SAME:** The value does not change.

***NOMAX:** There is no disconnect limit.

transmit-TDD-or-WACK-retry: Specify a value ranging from 0 through 65534 for the number of retries.

RCVRTY

Specifies the number of retries for receiving temporary text delay (TTD) or wait before transmitting (WACK) before indicating a session failure. This parameter applies only if the application type is *PGM.

***SAME:** The value does not change.

***NOMAX:** There is no disconnect limit.

receive-TTD-or-WACK-retry: Specify a value ranging from 0 through 65534 for the number of retries.

DSRDRPTMR

Specifies the amount of time the system waits for the modem to exit the Data Set Ready (DSR) state before signaling an error.

***SAME:** The value does not change.

DSR-drop-timer: Specify a value ranging from 3 through 60 seconds.

AUTOANSTYP

Specifies the method the system uses to answer incoming calls.

***SAME:** The value does not change.

***DTR:** The system enters the Data Terminal Ready state, signals the modem to answer calls, and waits for the modem to enter the Data Set Ready (DSR) state.

***CDSTL:** The system enters the Connect Data Set to Line (CDSTL) state after monitoring the Ring Indicator to signal the modem to answer the call.

CTSTMR

Specifies the amount of time the system waits for the modem to enter or exit the Clear to Send (CTS) state before signaling an error.

***SAME:** The value does not change.

CTS-timer: Specify a value ranging from 10 through 60 seconds.

RMTANSTMR

Specifies the amount of time the system waits for the modem to enter the Data Set Ready (DSR) state after dialing before signaling an error.

***SAME:** The value does not change.

answer-timer: Specify a value ranging from 30 through 120 seconds.

CMNRCYLMT

Specifies the number of recovery attempts made by the system before an inquiry message is sent to the system operator. Also specifies the time (in minutes) that must elapse before the system sends an inquiry message to the system operator indicating that the recovery attempt count limit is reached.

***SAME:** The value does not change.

***SYSVAL:** The recovery limits specified in the QCMNRCYLMT system value are used.

Element 1: Maximum Recovery Limit

count-limit: Specify the number of second-level recovery attempts to be made. Valid values range from 0 through 99.

Element 2: Recovery Time Interval

time-interval: Specify the time interval (in minutes) at which the specified number of second-level recoveries are attempted. Valid values range from 0 through 120. If the value specified for *count-limit* is not 0, the value 0 specifies infinite recovery.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

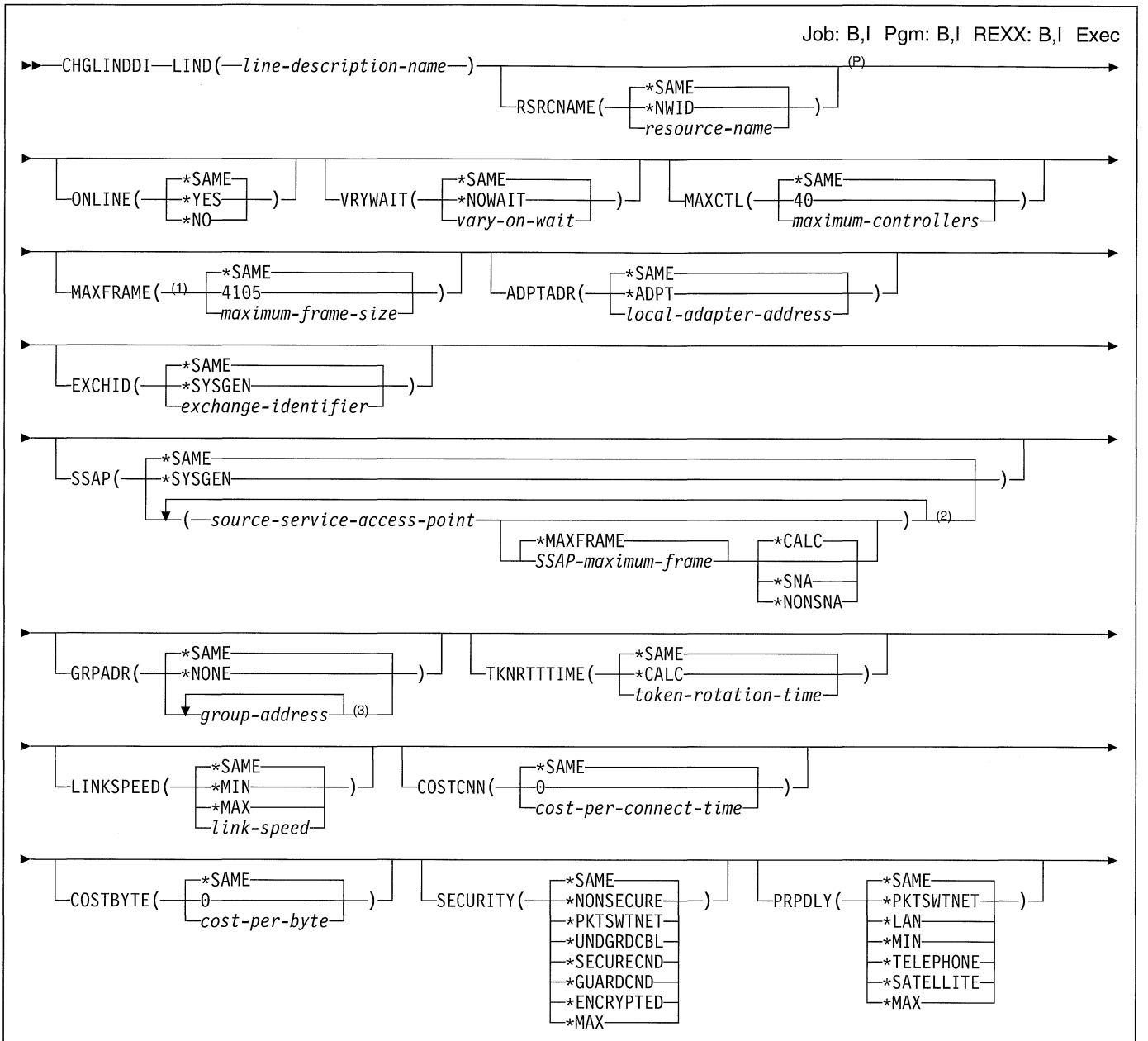
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGLINBSC  LIND(BSCLIN1)  INACTTMR(300)
```

This command changes the inactivity timer for line description BSCLIN1 to 300 (30.0 seconds).

CHGLINDDI (Change Line Description (DDI Network)) Command



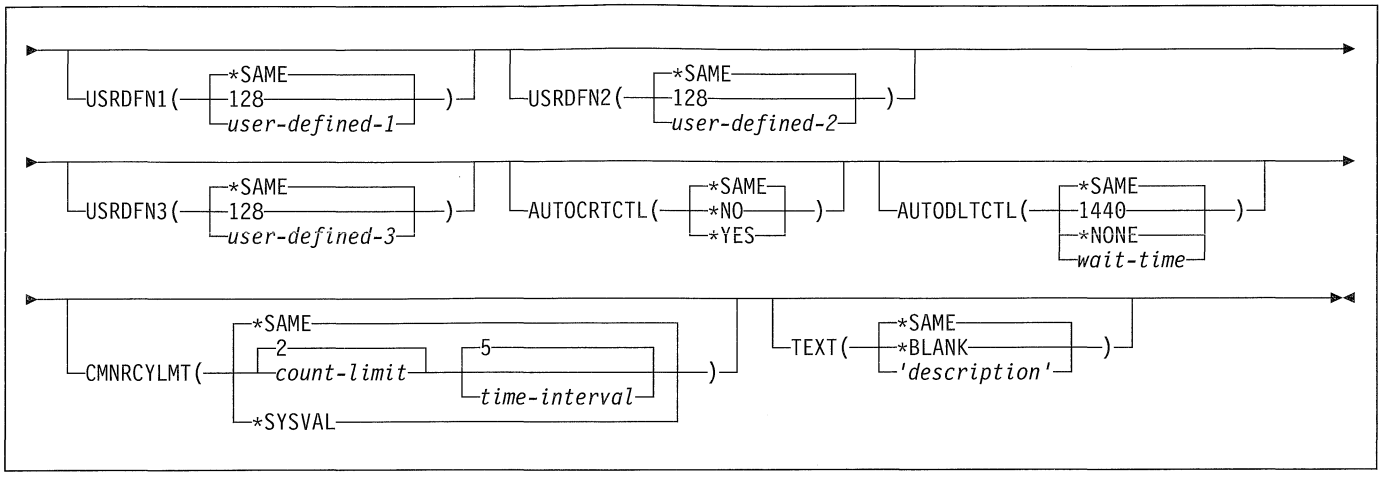
Notes:

P All parameters preceding this point can be specified positionally.

1 Any value ranging from 265 to 4444 can be specified.

2 A maximum of 24 repetitions

3 A maximum of 12 repetitions.



Purpose

The Change Line Description (Distributed Data Interface) (CHGLINDDI) command changes a line description for a distributed data interface line such as an FDDI (Fiber Distributed Data Interface).

Required Parameter

LIND

Specifies the name of the line description being changed.

Optional Parameters

RSRCNAME

Specifies the resource name that identifies the hardware that the description represents.

Note: Use the Work with Hardware Resources (WRKHDWRSC) command with *CMN specified on the TYPE parameter to help determine the resource name. Specify the resource name of the communications port. The resource name consists of the input/output adapter (IOA) resource name and the port number on the IOA. For example, if the resource name of the IOA is LIN01 and the port on the IOA is 1, then the resource name is LIN011.

The value specified on the RSRCNAME parameter cannot be changed from *NWID to another value or from another value to *NWID.

***SAME:** The value does not change.

***NWID:** The resource name is determined by the network interface used.

resource-name: Specify a resource name.

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The line is automatically varied on at IPL.

***NO:** This line is not automatically varied on at IPL.

VRYWAIT

Specifies whether the line is varied on asynchronously or synchronously. For synchronous vary on, specifies how long the system waits for the vary on to complete.

***SAME:** The value does not change.

***NOWAIT:** The system does not wait for the vary on to complete. The line is varied on asynchronously.

vary-on-wait: Specify the time (in seconds) to wait. Valid values range from 15 through 180. The system waits until the line is varied on, or until the specified time passes, before completing the Vary Configuration (VRYCFG) command.

Notes:

1. Specifying a wait time in the line description affects system IPL time, if ONLINE(*YES) is used, by the amount of time it takes to synchronously vary on the line or reach the wait-time value.
2. The time required to vary on a line is the time it takes to put tasks in place to manage the line, to activate the communications I/O processor (IOP) (including downloading the IOP model-unique Licensed Internal Code), and to establish communications with the data circuit-terminating equipment (DCE). Normal vary-on time ranges from 5 through 45 seconds, but can be longer, depending on the system, line protocol, and other factors.

MAXCTL

Specifies the maximum number of controllers that the line supports.

***SAME:** The value does not change.

40: The line supports 40 controllers.

maximum-controllers: Specify a number large enough to account for all controllers currently active to this network, and the controllers that will be attached in the near future. Valid values range from 1 through 256.

MAXFRAME

Specifies the maximum frame (path information unit (PIU)) size that the controller can send or receive. This value is used to calculate request unit (RU) sizes. Since the maximum PIU size that the controller can send or receive is negotiated at exchange identifier time, the maximum PIU size used at run time may be different. This value matches the corresponding value on the host system.

***SAME:** The value does not change.

4105: The maximum frame size is 4105 bytes.

maximum-frame-size: Specify a maximum frame size. Valid values range from 265 through 4444 bytes.

ADPTADR

Specifies the 12-character hexadecimal adapter address.

Note: ADPTADR(*ADPT) is not valid when RSRCNAME(*NWID) is specified.

***SAME:** The value does not change.

***ADPT:** This value gives the user the preset DDI default address for this DDI adapter card. The user may display this by doing a DSPLIND on this line description after it has successfully varied on.

local-adapter-address: Specify an address for this system in the DDI network. Valid values range from hexadecimal 400000000000 through 7FFFFFFFFFFF.

EXCHID

Specifies the hexadecimal exchange identifier that is used to identify the local system to the remote system. The 8-digit hexadecimal exchange identifier contains three digits for the block number and five digits for the identifier of this system.

***SAME:** The value does not change.

***SYSGEN:** The AS/400 system generates the exchange identifier.

exchange-identifier: Specify (if the *SYSGEN value is not specified) an exchange identifier composed of eight hexadecimal digits starting with 056.

SSAP

Specifies source service access points (SSAPs). This is the hexadecimal logical address used to route incoming data from the bus to the proper user. A maximum frame size can be specified for each SSAP. Valid SSAP values are 06 and AA (for TCP/IP), and 04 through 9C divisible by 4 (for SNA).

The destination service access point (DSAP), specified by the remote controller, must match one of the SSAPs specified in order for communication to occur. All SSAP values must be unique.

***SAME:** The value does not change.

***SYSGEN:** The system automatically creates three SSAPs, hex 04 for SNA, and hex AA and 06 for TCP/IP applications.

Element 1: SSAPs

source-service-access-point: Specify up to 24 SSAPs using valid SSAP values.

Element 2: Frame Size for SSAPs

***MAXFRAME:** The frame size specified on the MAXFRAME parameter is used.

SSAP-maximum-frame: Specify the maximum SSAP frame size (the maximum size of the data field that may be transmitted or received). Valid values for this parameter range from 265 through 4438 bytes.

Element 3: SSAP Type

***CALC:** The system determines the SSAP type based on the following values:

- 04 through 9C, divisible by 4 (for SNA)
- 02 through FE, divisible by 2 (for non-SNA)

***SNA:** The SSAP is used for SNA communications. Valid values range from 04 through 9C and must be divisible by 4.

***NONSNA:** The SSAP is used for non-SNA communications. Valid values range from 02 through FE and must be divisible by 2.

GRPADR

Specifies the distributed data interface group addresses used. Group addresses must each be specified as a 12-digit hexadecimal number. Valid values range from 800000000000 through FFFFFFFFFFFE.

***SAME:** The value does not change.

***NONE:** No group addresses are specified.

group-address: Specify a hexadecimal group address.

TKNRTTTIME

Specifies the token rotation time requested. This value is used when the station bids on the network. The lowest value of all attached stations on a ring determines the value the ring uses.

Note: TKNRTTTIME(*CALC) must be specified when RSRCNAME(*NWID) is specified.

***SAME:** The value does not change.

***CALC:** The system calculates the value based on the type of line that is linked to the controller.

token-rotation-time: Specify a value ranging from 4 through 167 milliseconds.

LINKSPEED

Specifies the link speed in bits per second (bps). This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

***MIN:** A link speed of less than 1200 bps is used.

***MAX:** A link speed greater than 16M bps is used.

link-speed: Specify the link speed. Valid values are: 1200, 2400, 4800, 7200, 9600, 14400, 19200, 48000,

56000, 64000, 112000, 128000, 168000, 192000, 224000, 256000, 280000, 320000, 336000, 384000, 448000, 499000, 576000, 614000, 691000, 768000, 845000, 922000, 998000, 1075000, 1152000, 1229000, 1382000, 1536000, 1690000, 1843000, 1997000, 4M, 10M, and 16M.

COSTCNN
Specifies the relative cost of being connected on the line. This parameter is required only if APPN is used on the system.

***SAME:** The value does not change.

cost-per-connect-time: Specify a value ranging from 0 through 255.

COSTBYTE
Specifies the relative cost per byte for sending and receiving data on the line. This parameter is required only if APPN is used on the system.

***SAME:** The value does not change.

cost-per-byte: Specify a value ranging from 0 through 255.

SECURITY
Specifies the security level of the physical line.

***SAME:** The value does not change.

***NONSECURE:** Normal priority is used.

***PKTSWTNET:** A packet switched network is used. Data does not always follow the same path through the network.

***UNDGRDCBL:** An underground cable is used.

***SECURECND:** A secure, unguarded conduit (for example, a pressurized pipe) is used.

***GUARDCND:** A guarded conduit, which is protected against physical tapping, is used.

***ENCRYPTED:** Data flowing on the line is encrypted.

***MAX:** A guarded conduit, protected against physical and radiation tapping, is used.

PRPDLY
Specifies the level of propagation delay on the line. This parameter is valid only if APPN is used on the system. The order of the values from shortest to longest delay is *MIN, *LAN, *TELEPHONE, *PKTSWTNET, and *SATELLITE.

***SAME:** The value does not change.

***LAN:** The local area network propagation delay is used.

***MIN:** The minimum propagation delay is used.

***TELEPHONE:** The telephone propagation delay is used.

***PKTSWTNET:** The packet switched network propagation delay is used.

***SATELLITE:** The satellite propagation delay is used.

***MAX:** The maximum propagation delay is used.

USRDFN1
Specifies the first of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

user-defined-1: Specify a value ranging from 0 through 255.

USRDFN2
Specifies the second of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

user-defined-2: Specify a value ranging from 0 through 255.

USRDFN3
Specifies the third of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

user-defined-3: Specify a value ranging from 0 through 255.

AUTOCRTCTL
Specifies whether the system automatically creates controller descriptions when calls are received from adjacent systems on the local area network (LAN).

***SAME:** The value does not change.

***NO:** The system does not automatically create a controller description when incoming calls are received.

***YES:** The system automatically creates a controller description when incoming calls are received.

AUTODLTCTL
Specifies the number of minutes an automatically created controller can remain in an idle state (switched from varied on to varied on pending) before the controller description and attached device descriptions are varied off and deleted.

***SAME:** The value does not change.

1440: The controller description can be idle for 1440 minutes (24 hours).

***NONE:** The system does not automatically delete or vary off the automatically configured, idle controller descriptions.

wait-time: Specify the number of minutes to wait before deleting the automatically configured, idle controller descriptions for this line. Valid values range from 1 to 10,000 minutes.

CMNRCYLMT

Specifies the number of recovery attempts made by the system before an inquiry message is sent to the system operator. Also specifies the time (in minutes) that must elapse before the system sends an inquiry message to the system operator indicating that the recovery attempt count limit is reached.

***SAME:** The value does not change.

Element 1: Maximum Recovery Limit

2: Two recovery attempts are made within the interval specified.

count-limit: Specify the number of recovery attempts to be made. Valid values range from 0 through 99.

Element 2: Recovery Time Interval

5: The specified number of recovery attempts is made within a 5-minute interval.

time-interval: Specify the time interval (in minutes) at which the specified number of second-level recoveries are attempted. Valid values range from 0 through 120. If the value specified for *count-limit* is not 0, the value 0 specifies infinite recovery.

Other Single Value

***SYSVAL:** The recovery limits specified in the QCMNRCYLMT system value are used.

TEXT

Specifies text that briefly describes the line description. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

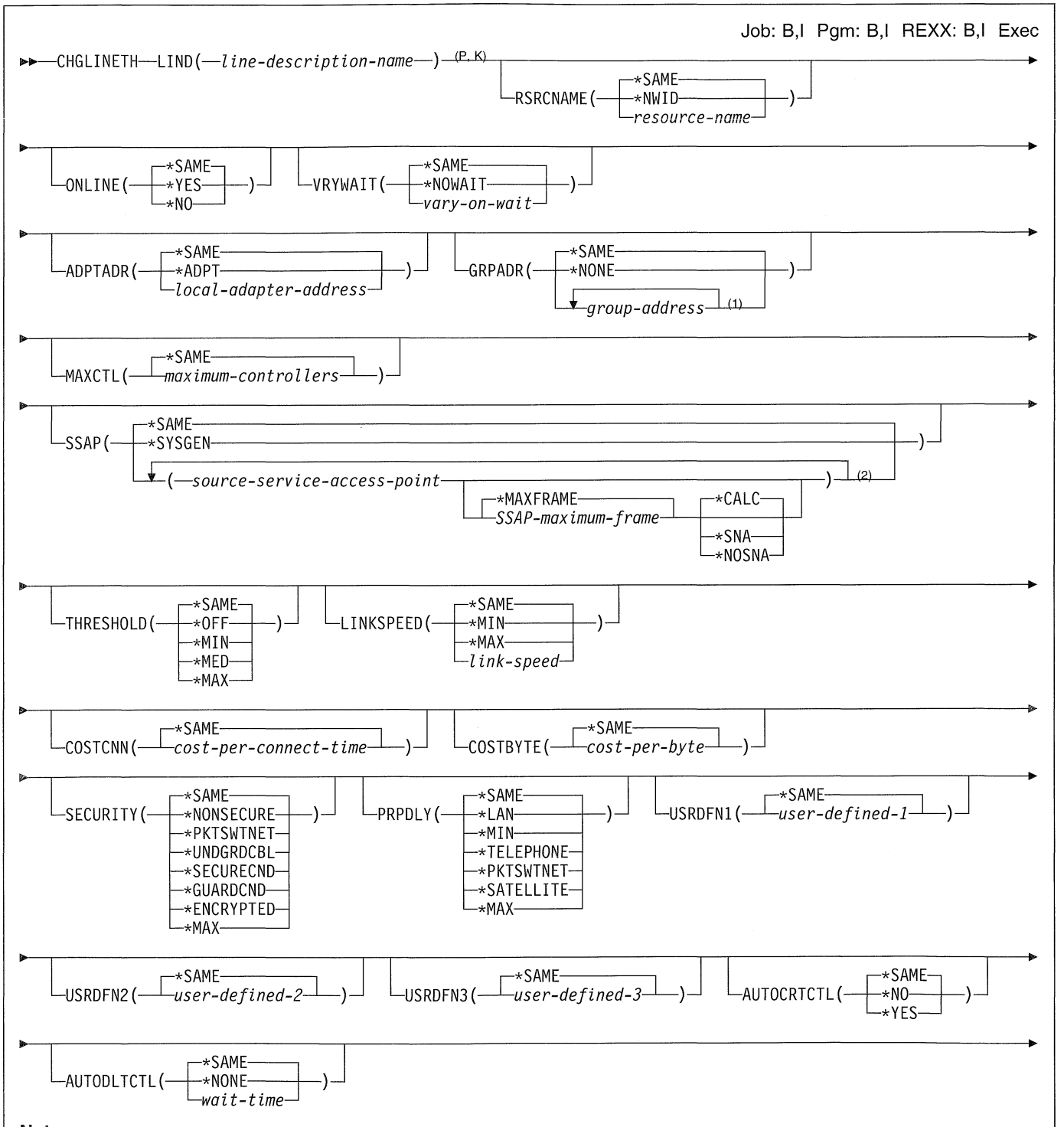
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGLINDDI   LIND(DDI1)  AUTODLTCTL(600)
  TEXT('NEW TEXT HERE')
```

This command changes DDI line DDI1 to automatically delete controllers after 600 seconds (10 minutes) and changes the text to 'NEW TEXT HERE'.

CHGLINETH (Change Line Description (Ethernet)) Command



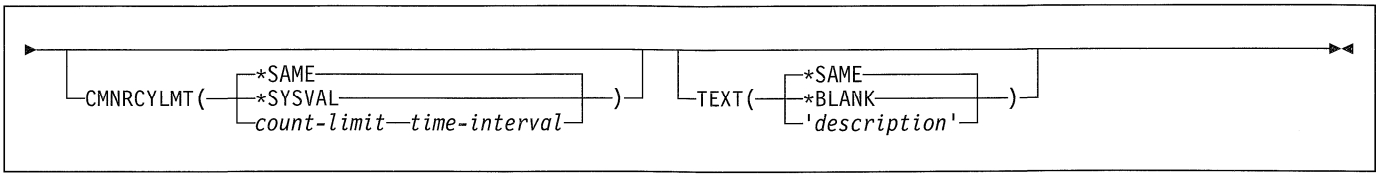
Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

1 A maximum of 12 repetitions

2 A maximum of 24 repetitions



Purpose

The Change Line Description (Ethernet) (CHGLINETH) command changes a line description for an Ethernet Local Area Network (LAN) line. More information about using this command is in the *OS/400* Communications Configuration Reference*.

Required Parameter

LIND

Specifies the name of the line description being changed.

Optional Parameters

RSRCNAME

Specifies the resource name of the communications port. The resource name consists of the input/output adapter (IOA) resource name and the port number on the IOA. For example, if the resource name of the IOA is LIN01 and the port on the IOA is 1, the resource name is LIN011.

Note: The value specified on the RSRCNAME parameter cannot be changed from *NWID to another value or from another value to *NWID.

***SAME:** The value does not change.

***NWID:** The resource name is determined by the network interface used.

resource-name: Specify the resource name of the communications port.

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The line is automatically varied on at IPL.

***NO:** This line is not automatically varied on at IPL.

VRYWAIT

Specifies whether the line is varied on asynchronously or synchronously. For synchronous vary on, specifies how long the system waits for the vary on to complete.

***SAME:** The value does not change.

***NOWAIT:** The system does not wait for the vary on to complete. The line is varied on asynchronously.

vary-on-wait: Specify the time (in seconds) to wait. Valid values range from 15 through 180. The system waits until the line is varied on, or until the specified time passes, before completing the Vary Configuration (VRYCFG) command.

Notes:

1. Specifying a wait time in the line description affects system IPL time, if ONLINE(*YES) is used, by the amount of time it takes to synchronously vary on the line or reach the wait-time value.
2. The time required to vary on a line is the time it takes to put tasks in place to manage the line, to activate the communications I/O processor (IOP) (including downloading the IOP model-unique Licensed Internal Code), and to establish communications with the data circuit-terminating equipment (DCE). Normal vary-on time ranges from 5 through 45 seconds, but can be longer, depending on the system, line protocol, and other factors.

ADPTADR

Specifies the 12-character hexadecimal adapter address.

Note: ADPTADR(*ADPT) is not valid when RSRCNAME(*NWID) is specified.

***SAME:** The value does not change.

***ADPT:** The preset Ethernet adapter address is used as the local adapter address. The adapter address can be displayed by using the DSPLIND command after the line description has been successfully varied on.

local-adapter-address: Specify the local Ethernet adapter address to override the preset local address. The local adapter address must be an individual address (it cannot be a group address). Valid values range from 020000000000 through FEFFFFFFFFFF in hexadecimal. The second digit (from the left) of the address must be 2, 6, A, or E.

GRPADR

Specifies whether an Ethernet group address is used.

***SAME:** The value does not change.

***NONE:** A group address is not used.

group-address: Specify the address to which a subset of units on the Ethernet line respond in addition to their local adapter address. Valid values range from 010000000000 through FDFFFFFFFFFF in hexadecimal form. All group addresses must be unique.

MAXCTL

Specifies the maximum number of controllers that the line supports.

***SAME:** The value does not change.

maximum-controllers: Specify the maximum number of controllers supported by the line, a number large enough to account for all of the controllers that are currently attached to this line, and for those controllers that will be attached in the near future. Valid values range from 1 through 256.

SSAP

Specifies source service access points (SSAPs). This is the hexadecimal logical address used to route incoming data from the Ethernet bus to the proper user. A maximum frame size can be specified for each SSAP. Valid SSAP values are 06 and AA (for TCP/IP), and 04 through 9C divisible by 4 (for SNA).

Note: Ethernet Version 2 (specified as *ETHV2 on the ETHSTD parameter) does not allow the SSAP values of 06 and AA.

The destination service access point (DSAP), specified by the remote controller, must match one of the SSAPs specified in order for communication to occur. All SSAP values must be unique.

***SAME:** The value does not change.

***SYSGEN:** For ETHSTD(*ALL or *IEEE8023), the AS/400 system creates three SSAPs: SSAP '04'X for SNA; 'AA'X and '06'X for TCP/IP applications. For ETHSTD(*ETHV2), the AS/400 system creates SSAP '04'X (for SNA).

Element 1: SSAPs

source-service-access-point: Specify up to 20 SSAPs, including hex AA and 06 for TCP/IP, and any hexadecimal number ranging from 04 through 9C that is divisible by 4 for SNA applications.

Element 2: Frame Size for SSAPs

***MAXFRAME:** The system determines the maximum frame size (data field size) that can be transmitted or received. If ETHSTD(*ALL or *IEEE8023) was specified, *CALC produces a frame size of 1496 for TCP/IP and SNA SSAPs. If ETHSTD(*ETHV2) was specified, *CALC produces a frame size of 1493 for SNA SSAPs.

SSAP-maximum-frame: Specify the maximum frame size for each SSAP.

Note: When RSRCNAME(*NWID) and ETHSTD(*ETHV2) are specified, the valid values for this parameter range from 265 through 1486 bytes. When RSRCNAME(*NWID) and ETHSTD(*ALL) or ETHSTD(*IEEE8023) are specified, the valid values for this parameter range from 265 through 1489 bytes.

Element 3: SSAP Type

***CALC:** The system determines the SSAP type based on the following values:

- 04 through 9C, divisible by 4 (for SNA)
- 02 through FE, divisible by 2 (for non-SNA)

***SNA:** The SSAP is used for SNA communications. Only SSAP values of 04 through 9C that are divisible by 4 are supported.

***NONSNA:** The SSAP is used for non-SNA Communications. Only SSAP values of 02 through FE and must be divisible by 2 are supported.

THRESHOLD

This parameter, and its values *SAME, *OFF, *MIN, *MED, and *MAX, can be specified but it is not used by the system starting in release V2R3M0. The parameter may be removed in a later release.

LINKSPEED

Specifies the link speed in bits per second (bps). This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

***MIN:** A link speed of less than 1200 bps is used.

***MAX:** A link speed greater than 16M bps is used.

link-speed: Specify the link speed. Valid values are: 1200, 2400, 4800, 7200, 9600, 14400, 19200, 48000, 56000, 64000, 112000, 128000, 168000, 192000, 224000, 256000, 280000, 320000, 336000, 384000, 448000, 499000, 576000, 614000, 691000, 768000, 845000, 922000, 998000, 1075000, 1152000, 1229000, 1382000, 1536000, 1690000, 1843000, 1997000, 4M, 10M, and 16M.

COSTCNN

Specifies the relative cost of being connected on the line. This parameter is required only if APPN is used on the system.

***SAME:** The value does not change.

cost-per-connect-time: Specify a value ranging from 0 through 255.

COSTBYTE

Specifies the relative cost per byte for sending and receiving data on the line. This parameter is required only if APPN is used on the system.

***SAME:** The value does not change.

cost-per-byte: Specify a value ranging from 0 through 255.

SECURITY

Specifies the security level of the physical line. This parameter is used only if APPN is used on the system.

***SAME:** The value does not change.

***NONSECURE:** Normal priority is used.

***PKTSWTNET:** A packet switched network is used. Data does not always follow the same path through the network.

***UNDGRDCBL:** An underground cable is used.

***SECURECND:** A secure, unguarded conduit (for example, a pressurized pipe) is used.

***GUARDCND:** A guarded conduit, which is protected against physical tapping, is used.

***ENCRYPTED:** Data flowing on the line is encrypted.

***MAX:** A guarded conduit, protected against physical and radiation tapping, is used.

PRPDLY
Specifies the level of propagation delay on the line. This parameter is valid only if APPN is used on the system. The order of the values from shortest to longest delay is *MIN, *LAN, *TELEPHONE, *PKTSWTNET, and *SATELLITE.

***SAME:** The value does not change.

***LAN:** The local area network propagation delay is used.

***MIN:** The minimum propagation delay is used.

***TELEPHONE:** The telephone propagation delay is used.

***PKTSWTNET:** The packet switched network propagation delay is used.

***SATELLITE:** The satellite propagation delay is used.

***MAX:** The maximum propagation delay is used.

USRDFN1
Specifies the first of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

user-defined-1: Specify a value ranging from 0 through 255.

USRDFN2
Specifies the second of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

user-defined-2: Specify a value ranging from 0 through 255.

USRDFN3
Specifies the third of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

user-defined-3: Specify a value ranging from 0 through 255.

AUTOCRTCTL
Specifies whether the system will automatically create controller descriptions when calls are received from adjacent systems on the local area network (LAN).

***SAME:** The value does not change.

***NO:** The system will not automatically create a controller description when incoming calls are received.

***YES:** The system will automatically create a controller description when incoming calls are received.

AUTODLTCTL
Specifies the number of minutes an automatically created controller can remain in an idle state (switched from varied on to varied on pending) before the controller description and attached device descriptions are varied off and deleted.

***SAME:** The value does not change.

***NONE:** The system will not automatically delete or vary off the automatically configured, idle controller descriptions.

wait-time: Specify the number of minutes to wait before deleting the automatically configured, idle controller descriptions for this line. Valid values range from 1 to 10,000 minutes.

CMNRCYLMT
Specifies the number of recovery attempts made by the system before an inquiry message is sent to the system operator. Also specifies the time (in minutes) that must elapse before the system sends an inquiry message to the system operator indicating that the recovery attempt count limit is reached.

***SAME:** The value does not change.

***SYSVAL:** The recovery limits specified in the QCMNRCYLMT system value are used.

Element 1: Maximum Recovery Limit

count-limit: Specify the number of recovery attempts to be made. Valid values range from 0 through 99.

Element 2: Recovery Time Interval

time-interval: Specify the time interval (in minutes) at which the specified number of second-level recoveries are attempted. Valid values range from 0 through 120. If the value specified for *count-limit* is not 0, the value 0 specifies infinite recovery.

TEXT
Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

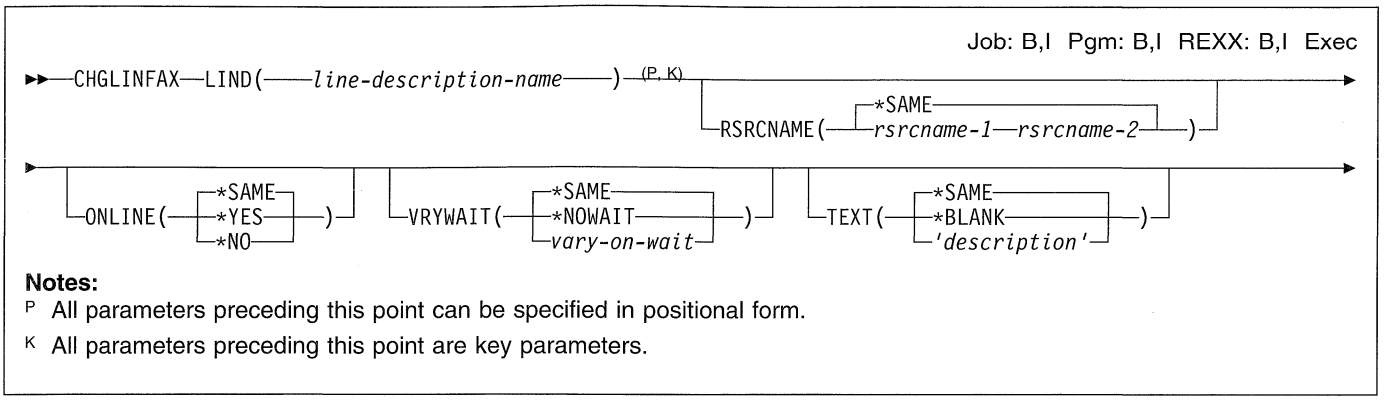
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGLINETH  LIND(BOSTON)  RSRCNAME(LIN041)
```

This command changes the resource name of an Ethernet line description named BOSTON to LIN041.

CHGLINFAX (Change Line Description (Fax)) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Line Description (Fax) (CHGLINFAX) command changes a line description for a facsimile (fax) line.

Required Parameter

LIND

Specifies the name of the line description being changed.

Optional Parameters

RSRCNAME

Specifies the resource names that describe the fax ports.

Note: Use the Work with Hardware Resources (WRKHDWRSC) command with *CMN specified for the TYPE parameter to help determine the resource name.

The resource name consists of the input/output adapter (IOA) resource name and the port number on the IOA. For example, if the resource name of the IOA is LIN01, the resource names for ports 1 and 2 are is LIN011 and LIN012.

The resource name for both ports of the fax IOA must be specified. All lines specified must be attached to the same input/output processor.

***SAME:** The value does not change.

rsrcname-1: Specify the first resource name to be used to describe the fax ports.

rsrcname-2: Specify the second resource name to be used to describe the fax ports.

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The line is automatically varied on at IPL.

***NO:** This line is not automatically varied on at IPL.

VRYWAIT

Specifies whether the line is varied on asynchronously or synchronously. For synchronous vary on, specifies how long the system waits for the vary on to complete.

***SAME:** The value does not change.

***NOWAIT:** The system does not wait for the vary on to complete. The line is varied on asynchronously.

vary-on-wait: Specify the time (in seconds) to wait. Valid values range from 15 through 180. The system waits until the line is varied on, or until the specified time passes, before completing the Vary Configuration (VRYCFG) command.

Notes:

1. Specifying a wait time in the line description affects system IPL time, if ONLINE(*YES) is used, by the amount of time it takes to synchronously vary on the line or reach the wait-time value.
2. The time required to vary on a line is the time it takes to put tasks in place to manage the line, to activate the communications I/O processor (IOP) (including downloading the IOP model-unique Licensed Internal Code), and to establish communications with the data circuit-terminating equipment (DCE). Normal vary-on time ranges from 5 through 45 seconds, but can be longer, depending on the system, line protocol, and other factors.

TEXT

Specifies text that briefly describes the line description. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

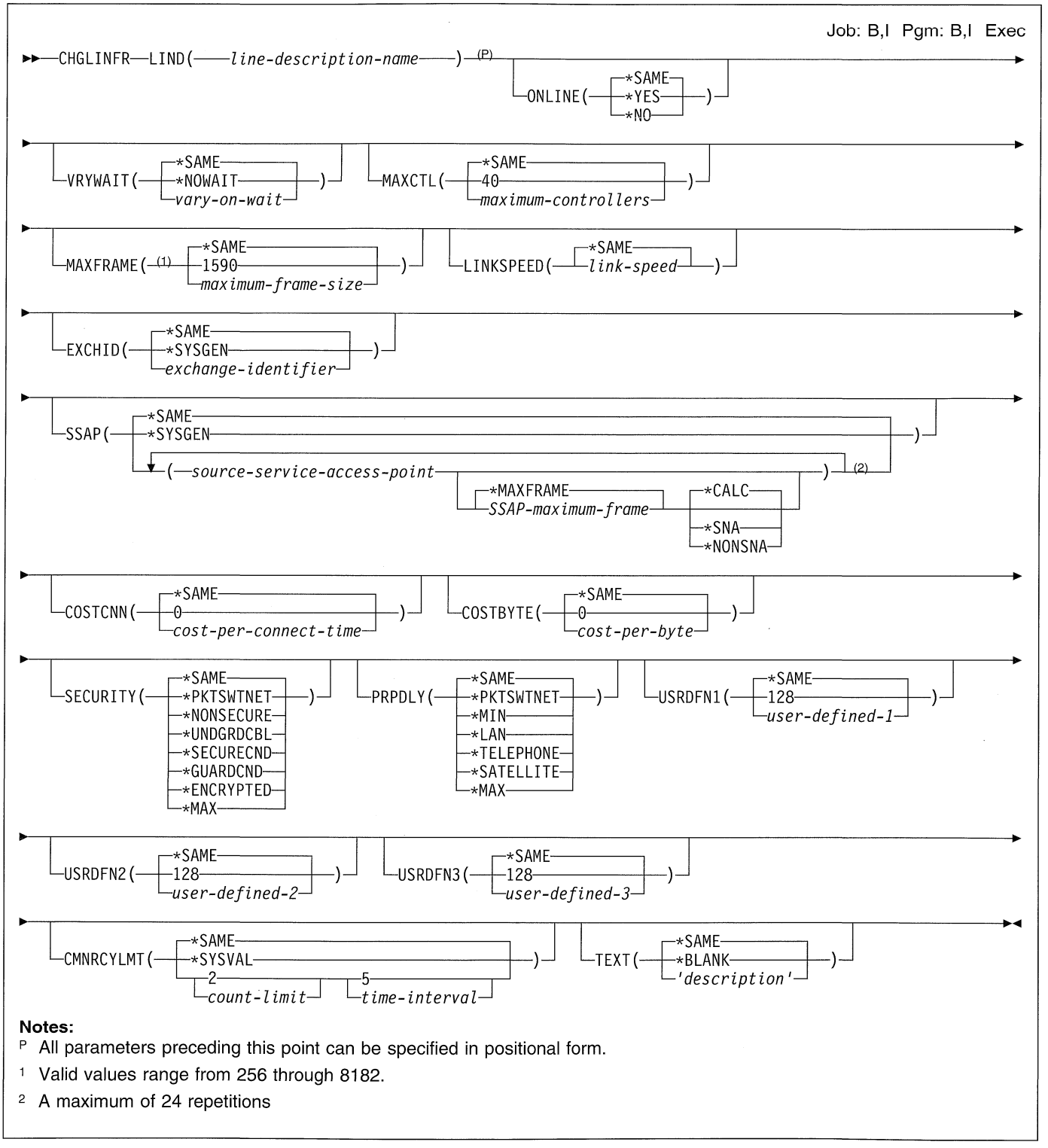
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGLINFAX   LIND(FAXLINE)  RSRCNAME(LIN031 LIN032)
```

This command changes the resource names for fax line description FAXLINE to LIN041 and LIN042.

CHGLINFR (Change Line Description (Frame Relay Network)) Command



Notes:

P All parameters preceding this point can be specified in positional form.

1 Valid values range from 256 through 8182.

2 A maximum of 24 repetitions

Purpose

The Change Line Description (Frame-Relay Network) (CHGLINFR) command changes a line description for a frame-relay network (FR) line. More information about using this command is in the *OS/400* Communications Configuration Reference.*

Required Parameter

LIND

Specifies the name of the line description being changed.

line-description-name: Specify the name of a line description.

Optional Parameters

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The line is automatically varied on at IPL.

***NO:** This line is not automatically varied on at IPL.

VRYWAIT

Specifies whether the line is varied on asynchronously or synchronously. For synchronous vary on, specifies how long the system waits for the vary on to complete.

***SAME:** The value does not change.

***NOWAIT:** The system does not wait for the vary on to complete. The line is varied on asynchronously.

vary-on-wait: Specify the time (in seconds) to wait. Valid values range from 15 through 180. The system waits until the line is varied on, or until the specified time passes, before completing the Vary Configuration (VRYCFG) command.

Notes:

1. Specifying a wait time in the line description affects system IPL time, if ONLINE(*YES) is used, by the amount of time it takes to synchronously vary on the line or reach the wait-time value.
2. The time required to vary on a line is the time it takes to put tasks in place to manage the line, to activate the communications I/O processor (IOP) (including downloading the IOP model-unique Licensed Internal Code), and to establish communications with the data circuit-terminating equipment (DCE). Normal vary-on time ranges from 5 through 45 seconds, but can be longer, depending on the system, line protocol, and other factors.

MAXCTL

Specifies the maximum number of controllers that the line supports.

***SAME:** The value does not change.

40: The line supports 40 controllers.

maximum-controllers: Specify a number large enough to account for all controllers currently active to this network, and the controllers that will be attached in the near future. Valid values range from 1 through 256.

MAXFRAME

Specifies the maximum frame (path information unit (PIU)) size that the controller can send or receive. This value is used to calculate request unit (RU) sizes. Since the maximum PIU size that the controller can send or receive is negotiated at exchange identifier time, the maximum PIU size used at run time may be different. This value matches the corresponding value on the host system. The recommended MAXFRAME values are: 502, 1014, 1590, 2038, 4086, and 8182 bytes.

***SAME:** The value does not change.

1590: The maximum frame size is 1590.

maximum-frame-size: Specify the maximum frame size value to be used. Valid values range from 265 through 8182.

LINKSPEED

Specifies the link speed in bits per second (bps). This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

link-speed: Specify the link speed. Valid values are: 1200, 2400, 4800, 7200, 9600, 14400, 19200, 48000, 56000, 64000, 112000, 128000, 168000, 192000, 224000, 256000, 280000, 320000, 336000, 384000, 448000, 499000, 576000, 614000, 691000, 768000, 845000, 922000, 998000, 1075000, 1152000, 1229000, 1382000, 1536000, 1690000, 1843000, 1997000.

EXCHID

Specifies the hexadecimal exchange identifier that is used to identify the local system to the remote system. The 8-digit hexadecimal exchange identifier contains three digits for the block number and five digits for the identifier of this system.

***SAME:** The value does not change.

***SYSGEN:** The AS/400 system generates the exchange identifier.

exchange-identifier: Specify (if the *SYSGEN value is not specified) an exchange identifier composed of eight hexadecimal digits starting with 056.

SSAP

Specifies the source service access point (SSAP). The most commonly used SNA SSAP is hex 04. All SSAP values must be unique.

***SAME:** The value does not change.

***SYSGEN:** The system automatically creates three SSAPs, hex 04 for SNA, and hex AA and 06 for TCP/IP applications.

Element 1: SSAPs

source-service-access-point: Specify up to 24 SSAPs, including hex AA and 06 for TCP/IP, and any hexadecimal number ranging from 04 through 9C that is divisible by four for SNA applications.

Element 2: Frame Size for SSAPs

***MAXFRAME:** The frame size specified on the MAXFRAME parameter is used.

SSAP-maximum-frame: Specify the maximum SSAP frame size (the maximum size of the data field that may be transmitted or received). Valid values for this parameter range from 265 through 8182 bytes.

Element 3: SSAP Type

***CALC:** The system determines the SSAP type based on the following values:

- 04 through 9C, divisible by 4 (for SNA)
- 02 through FE, divisible by 2 (for non-SNA)

***SNA:** The SSAP is used for SNA communications. Valid values range from 04 through 9C and must be divisible by 4.

***NONSNA:** The SSAP is used for non-SNA communications. Valid values range from 02 through FE and must be divisible by 2.

COSTCNN

Specifies the relative cost of being connected on the line. This parameter is required only if APPN is used on the system.

***SAME:** The value does not change.

0: The cost per connect time is 0.

cost-per-connect-time: Specify a value ranging from 0 through 255.

COSTBYTE

Specifies the relative cost per byte for sending and receiving data on the line. This parameter is required only if APPN is used on the system.

***SAME:** The value does not change.

0: The cost per byte is 0.

cost-per-byte: Specify a value ranging from 0 through 255.

SECURITY

Specifies the security level of the physical line. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

***PKTSWTNET:** A packet switched network is used. Data does not always follow the same path through the network.

***NONSECURE:** Normal priority is used.

***UNDGRDCBL:** An underground cable is used.

***SECURECND:** A secure, unguarded conduit (for example, a pressurized pipe) is used.

***GUARDCND:** A guarded conduit, which is protected against physical tapping, is used.

***ENCRYPTED:** Data flowing on the line is encrypted.

***MAX:** A guarded conduit, protected against physical and radiation tapping, is used.

PRPDLY

Specifies the level of propagation delay on the line. This parameter is valid only if APPN is used on the system. The order of the values from shortest to longest delay is *MIN, *LAN, *TELEPHONE, *PKTSWTNET, and *SATELLITE.

***SAME:** The value does not change.

***PKTSWTNET:** The packet switched network propagation delay is used.

***MIN:** The minimum propagation delay is used.

***LAN:** The local area network propagation delay is used.

***TELEPHONE:** The telephone propagation delay is used.

***SATELLITE:** The satellite propagation delay is used.

***MAX:** The maximum propagation delay is used.

USRDFN1

Specifies the first of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

128: The value 128 is used.

user-defined-1: Specify a value ranging from 0 through 255.

USRDFN2

Specifies the second of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

128: The value 128 is used.

user-defined-2: Specify a value ranging from 0 through 255.

USRDFN3

Specifies the third of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

128: The value 128 is used.

user-defined-3: Specify a value ranging from 0 through 255.

CMNRCYLMT

Specifies the number of recovery attempts made by the system before an inquiry message is sent to the system operator. Also specifies the time (in minutes) that must elapse before the system sends an inquiry message to the system operator indicating that the recovery attempt count limit is reached.

***SAME:** The value does not change.

***SYSVAL:** The recovery limits specified in the QCMNRCYLMT system value are used.

Element 1: Maximum Recovery Limit

2: Two recovery attempts are made within the interval specified.

count-limit: Specify the number of recovery attempts to be made. Valid values range from 0 through 99.

Element 2: Recovery Time Interval

5: The specified number of recovery attempts is made within a 5-minute interval.

time-interval: Specify the time interval (in minutes) at which the specified number of second-level recoveries are attempted. Valid values range from 0 through 120. If the value specified for *count-limit* is not 0, the value 0 specifies infinite recovery.

TEXT

Specifies text that briefly describes the line description. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

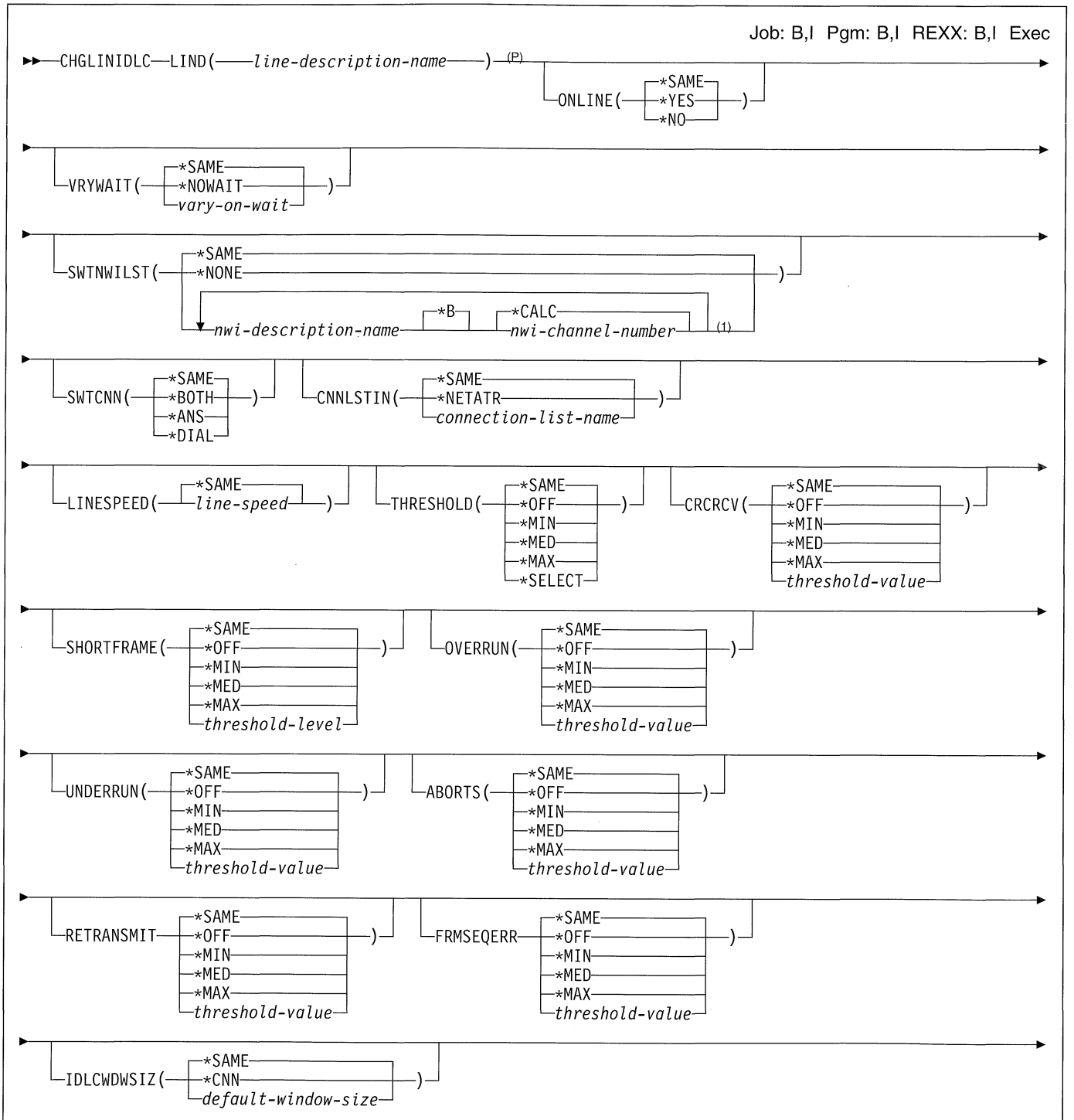
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGLINFR   LIND(FR2)  ONLINE(*NO)
  MAXFRAME(4444)  LINKSPEED(1536000)
```

This command changes frame relay line FR2 so it does not vary on automatically at initial program load (IPL). The maximum frame size is set to 4444 bytes and the link speed is set to 1536000 bits per second (bps).

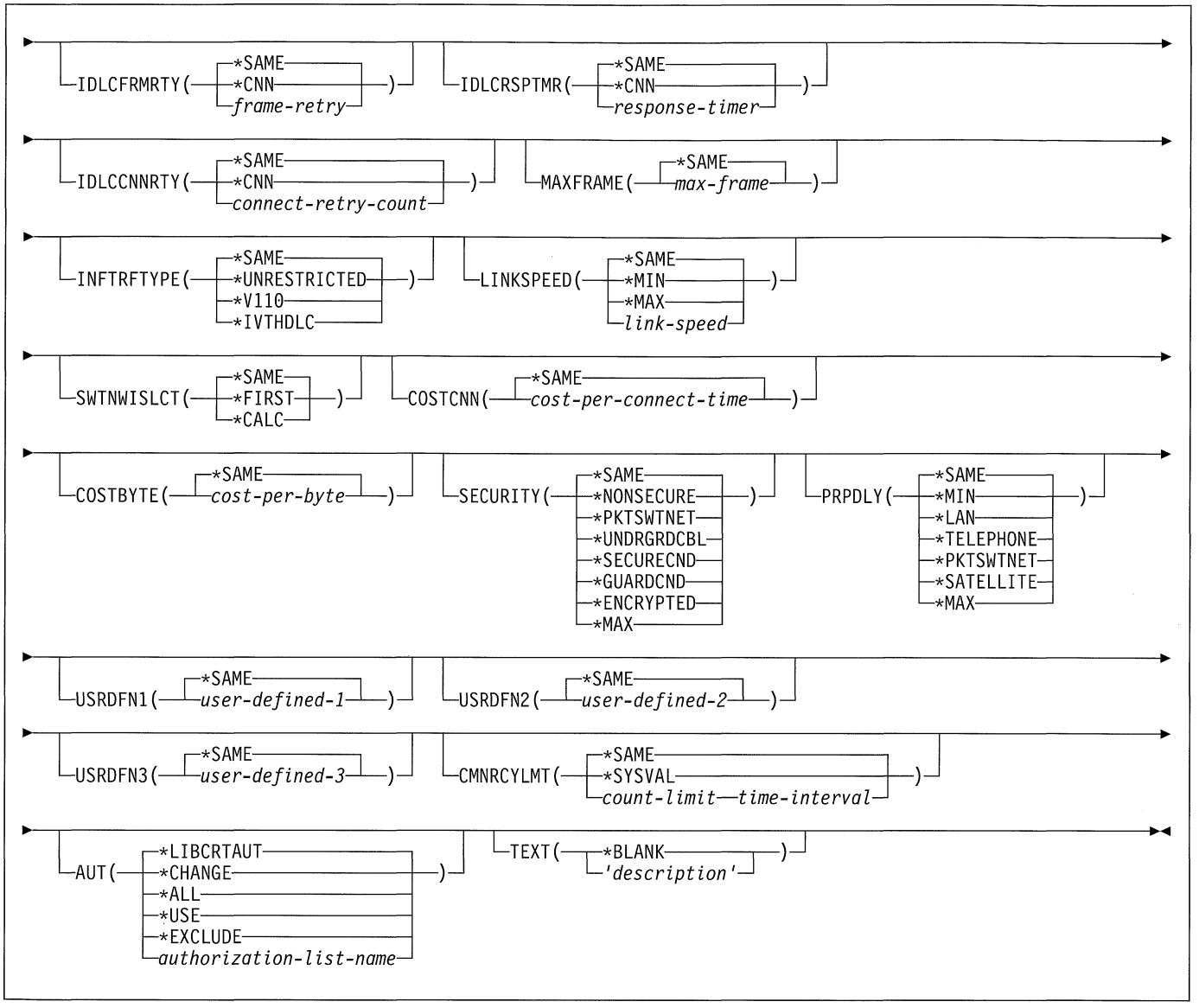
CHGLINIDLC (Change Line Description (IDLC)) Command



Notes:

P All parameters preceding this point can be specified in positional form.

1 A maximum of 64 repetitions



Purpose

The Change Line Description (IDLC) (CHGLINIDLC) command changes a line description for an Integrated Services Digital Network (ISDN) Data Link Controller (IDLC) line.

Required Parameter

LIND

Specifies the name of the line description being changed.

Optional Parameters

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The line is automatically varied on at IPL.

***NO:** This line is not automatically varied on at IPL.

VRYWAIT

Specifies whether the line is varied on asynchronously or synchronously. For synchronous vary on, specifies how long the system waits for the vary on to complete.

***SAME:** The value does not change.

***NOWAIT:** The system does not wait for the vary on to complete. The line is varied on asynchronously.

vary-on-wait: Specify the time (in seconds) to wait. Valid values range from 15 through 180. The system waits until the line is varied on, or until the specified time passes, before completing the Vary Configuration (VRYCFG) command.

Notes:

1. Specifying a wait time in the line description affects system IPL time, if ONLINE(*YES) is used, by the amount of time it takes to synchronously vary on the line or reach the wait-time value.
2. The time required to vary on a line is the time it takes to put tasks in place to manage the line, to activate the communications I/O processor (IOP) (including downloading the IOP model-unique Licensed Internal Code), and to establish communications with the data circuit-terminating equipment (DCE). Normal vary-on time ranges from 5 through 45 seconds, but can be longer, depending on the system, line protocol, and other factors.

SWTNWILST

Specifies, for ISDN switched connections, a list of network interface descriptions to which this line can be attached. The first available network description is chosen from the list when an incoming or outgoing call is processed.

***SAME:** The list does not change.

***NONE:** A network interface description is not specified for this line.

Element 1: Network Interface Description Name

nwi-description-name: Specify, for switched connections, the name of the network interface description to which this line attaches.

Element 2: Network Interface Channel Type

***B:** The B channel is used.

Element 3: Network Interface Channel-Number

***CALC:** The system selects the channel number (based on availability) defined for the network interface description when an incoming or outgoing call is processed.

nwi-channel-number: Specify the channel number. Valid values are 1 and 2.

SWTCNN

Specifies whether the switched line is used for incoming calls, outgoing calls, or both.

***SAME:** The value does not change.

***BOTH:** The line is used for both incoming and outgoing calls.

***ANS:** The line is used for incoming calls only.

***DIAL:** The line is used for outgoing calls only.

CNNLSTIN

Specifies the name of the connection list that is used to retrieve call or connection information for identifying authorized incoming calls.

***SAME:** The value does not change.

***NETATR:** The connection list used by this line description is taken from the list of system default network attributes that were identified at IPL (Initial Program Load). Use the Display Network Attributes (DSPNETA) command to see the name of the default ISDN connection list (DFTCNNLST) parameter that is used.

connection-list-name: Specify the name of the connection list used for this line description.

LINESPEED

Specifies the line speed in bits per second (bps).

***SAME:** The value does not change.

line-speed: Specify the line speed. Valid values are: 56000 and 64000 bps.

THRESHOLD

Specifies the threshold for the number of errors beyond which a message is sent informing the user that errors have occurred. All error threshold parameters are set to the value specified, unless *SELECT is specified, in which case each error threshold parameter is set individually. The next five parameters are the threshold parameters. They are: CRCRCV, OVERRUN, UNDERRUN, ABORTS, and SHORTFRAME.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** The error threshold is set at a minimum monitoring level.

***MED:** The system performs a medium amount of error threshold monitoring for all types of errors.

***MAX:** The error threshold is set at a maximum monitoring level.

***SELECT:** The threshold for the number of errors beyond which a message is sent informing the user that errors have occurred is set individually for each parameter.

CRCRCV

Specifies the level of error threshold monitoring done by the system for Cyclic Redundancy Check (CRC) errors.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** The system performs a minimum amount of error threshold monitoring for CRC errors: 6 errors in the first 30 seconds or 180 errors in any 15 minutes. The system is more tolerant of errors than if *MED or *MAX is specified.

***MED:** The system performs a medium amount of error threshold monitoring for CRC errors: 2 errors in the first 30 seconds or 60 errors in any 15 minutes.

***MAX:** The system performs the maximum amount of error threshold monitoring for CRC errors received.

threshold-value: Specify a value ranging from 1 through 10000. The number represents the number of errors received in a 15-minute interval.

SHORTFRAME

Specifies the threshold for the level of errors for short frame errors received.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** A minimum of monitoring is done: 6 X.25 or SDLC frames received in the first 30 seconds or 1 received every second for 14 minutes. The system is more tolerant of errors than if *MED or *MAX is specified.

***MED:** A medium amount of monitoring is done: 3 X.25 or SDLC frames received in the first 30 seconds or 1 received every 3 seconds for 10-14 minutes.

***MAX:** A maximum amount of monitoring is done.

threshold-level: Specify a threshold level. Valid values range from 1 through 10000. The number represents the number of errors received in a 15-minute interval.

OVERRUN

Specifies the level of error threshold monitoring done by the system for overrun errors.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** The system performs a minimum amount of error threshold monitoring for buffer overrun errors: 2 errors in the first 90 seconds or 20 errors in any 15 minutes. The system is more tolerant of errors than if *MED or *MAX is specified.

***MED:** The system performs a medium amount of error threshold monitoring for buffer overrun errors: 2 errors in the first 300 seconds or 6 errors in any 15 minutes.

***MAX:** The system performs the maximum amount of error threshold monitoring for buffer overrun errors.

threshold-value: Specify a value ranging from 1 through 3000. The number represents the number of errors received in a 15-minute interval.

UNDERRUN

Specifies the level of error threshold monitoring done by the system for buffer underrun errors.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** The system performs a minimum amount of error threshold monitoring for buffer overrun errors: 2 errors in the first 90 seconds or 20 errors in any 15 minutes. The system is more tolerant of errors than if *MED or *MAX is specified.

***MED:** The system performs a medium amount of error threshold monitoring for buffer overrun errors: 2 errors in the first 300 seconds or 6 errors in any 15 minutes.

***MAX:** The system performs the maximum amount of error threshold monitoring for buffer underrun errors.

threshold-value: Specify a value ranging from 1 through 3000. The number represents the number of errors received in a 15-minute interval.

ABORTS

Specifies the level of error threshold monitoring done by the system for aborted frames.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** The system performs a minimum amount of error threshold monitoring for aborted frames received: 6 errors in the first 30 seconds or 180 errors in any 15 minutes. The system is more tolerant of errors than if *MED or *MAX is specified.

***MED:** The system performs a medium amount of error threshold monitoring for aborted frames received: 2 errors in the first 30 seconds or 60 errors in any 15 minutes.

***MAX:** The system performs the maximum amount of error threshold monitoring for aborted frames received.

threshold-value: Specify a value ranging from 1 through 5000. The number represents the number of errors received in a 15-minute interval.

RETRANSMIT

Specifies the threshold for the number of frame retransmissions beyond which a message is sent informing the user that errors have occurred.

***SAME:** The value does not change.

***OFF:** Error threshold monitoring is turned off for retransmitted frames.

***MIN:** The system performs the minimum amount of error threshold monitoring for retransmitted frames: 5 errors in the first 30 seconds or 150 errors in any 15 minutes. The system is more tolerant of errors than if *MED or *MAX is specified.

***MED:** The system performs a medium amount of error threshold monitoring for retransmitted frames: 2 errors in the first 60 seconds or 30 errors in any 15 minutes.

***MAX:** The error threshold is set at a maximum monitoring level.

threshold-value: Specify a number, ranging from 1 through 10000, that corresponds to the number of errors allowed in a 15-minute interval.

FRMSEQERR

Specifies the threshold for the number of frame sequence errors beyond which a message is sent informing the user that errors have occurred.

***SAME:** The value does not change.

***OFF:** Error threshold monitoring is turned off for frame sequence errors.

***MIN:** The system performs the minimum amount of error threshold monitoring for frame sequence errors: 2 errors in the first 90 seconds or 20 errors in any 15

minutes. The system is more tolerant of errors than if *MED or *MAX is specified.

***MED:** The system performs a medium amount of error threshold monitoring for frame sequence errors: 2 errors in the first 300 seconds or 6 errors in any 15 minutes.

***MAX:** The system performs the maximum amount of error threshold monitoring for frame sequence errors.

threshold-value: Specify a number, ranging from 1 through 3000, that corresponds to the number of errors allowed in a 15-minute interval.

IDLCWDWSIZ

Specifies the default window size.

***SAME:** The value does not change.

***CNN:** The value is determined according to the value specified on the CNN parameter.

default-window-size: Specify a value ranging from 1 through 31.

IDLCFRMRTY

Specifies the maximum number of retransmissions before reporting an error.

***SAME:** The value does not change.

***CNN:** The value is determined according to the value specified on the CNN parameter.

frame-retry: Specify a value ranging from 0 through 100.

IDLCRSPTMR

Specifies the amount of time to wait before retransmitting a frame when an acknowledgement has not been received.

***SAME:** The value does not change.

***CNN:** The value is determined according to the value specified on the CNN parameter.

response-timer: Specify a value ranging from 10 through 100 tenths of seconds. For example, 100 tenths of seconds equals 10 seconds.

IDLCCNNRTY

Specifies the number of times a transmission can be retried at connection time. Specifying *NOMAX indicates to try until successful.

***SAME:** The value does not change.

***CNN:** The value is determined according to the value specified on the CNN parameter.

***NOMAX:** There is no disconnect limit.

connect-retry-count: Specify a value ranging from 1 through 100.

MAXFRAME

Specifies the maximum frame size that can be transmitted and received on this line description.

***SAME:** The value does not change.

max-frame: Specify a value ranging from 265 through 8196.

INFTRFTYPE

Specifies the information transfer type. The information transfer type determines the layer-1 protocol.

***SAME:** The value does not change.

***UNRESTRICTED:** The data-channel traffic appears as digital information; no physical transformation is required and each B-channel operates at capacity (64k bits per second (bps)).

***V110:** The transfer type is V-series Recommendation 110. Each B-channel operates at 56k bps.

***IVTHDLC:** The transfer type is Inverted HDLC. Each B-channel operates at capacity, 64k bps.

LINKSPEED

Specifies the link speed in bits per second (bps). This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

***MIN:** A link speed of less than 1200 bps is used.

***MAX:** A link speed greater than 16M bps is used.

link-speed: Specify the link speed. Valid values are: 1200, 2400, 4800, 7200, 9600, 14400, 19200, 48000, 56000, 64000, 112000, 128000, 168000, 192000, 224000, 256000, 280000, 320000, 336000, 384000, 448000, 499000, 576000, 614000, 691000, 768000, 845000, 922000, 998000, 1075000, 1152000, 1229000, 1382000, 1536000, 1690000, 1843000, 1997000, 2048000, 4M, 10M, and 16M.

SWTNWISLCT

Specifies the method used to select network interfaces from the switched network interface list.

***SAME:** The value does not change.

***FIRST:** Selection begins with the first network interface specified in the switched network interface list.

***CALC:** The system calculates which network interface is selected.

COSTCNN

Specifies the relative cost of being connected on the line. This parameter is required only if APPN is used on the system.

***SAME:** The value does not change.

cost-per-connect-time: Specify a value ranging from 0 through 255.

COSTBYTE

Specifies the relative cost per byte for sending and receiving data on the line. This parameter is required only if APPN is used on the system.

***SAME:** The value does not change.

cost-per-byte: Specify a value ranging from 0 through 255.

SECURITY

Specifies the security level of the physical line. This parameter is used only if APPN is used on the system.

***SAME:** The value does not change.

***NONSECURE:** Normal priority is used.

***PKTSWTNET:** A packet switched network is used. Data does not always follow the same path through the network.

***UNDGRDCBL:** An underground cable is used.

***SECURECND:** A secure, unguarded conduit (for example, a pressurized pipe) is used.

***GUARDCND:** A guarded conduit, which is protected against physical tapping, is used.

***ENCRYPTED:** Data flowing on the line is encrypted.

***MAX:** A guarded conduit, protected against physical and radiation tapping, is used.

PRPDLY

Specifies the level of propagation delay on the line. This parameter is valid only if APPN is used on the system. The order of the values from shortest to longest delay is *MIN, *LAN, *TELEPHONE, *PKTSWTNET, and *SATELLITE.

***SAME:** The value does not change.

***MIN:** The minimum propagation delay is used.

***LAN:** The local area network propagation delay is used.

***TELEPHONE:** The telephone propagation delay is used.

***PKTSWTNET:** The packet switched network propagation delay is used.

***SATELLITE:** The satellite propagation delay is used.

***MAX:** The maximum propagation delay is used.

USRDFN1

Specifies the first of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

user-defined-1: Specify a value ranging from 0 through 255.

USRDFN2

Specifies the second of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

user-defined-2: Specify a value ranging from 0 through 255.

USRDFN3

Specifies the third of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

user-defined-3: Specify a value ranging from 0 through 255.

CMNRCYLMT

Specifies the number of recovery attempts made by the system before an inquiry message is sent to the system operator. Also specifies the time (in minutes) that must elapse before the system sends an inquiry message to the system operator indicating that the recovery attempt count limit is reached.

***SAME:** The value does not change.

***SYSVAL:** The recovery limits specified in the QCMNRCYLMT system value are used.

Element 1: Maximum Recovery Limit

count-limit: Specify the number of recovery attempts to be made. Valid values range from 0 through 99.

Element 2: Recovery Time Interval

time-interval: Specify the time interval (in minutes) at which the specified number of second-level recoveries are attempted. Valid values range from 0 through 120. If the value specified for *count-limit* is not 0, the value 0 specifies infinite recovery.

AUT

Specifies the authority given to users who do not have specific authority to the object, who are not on the authorization list, and whose user group has no specific authority to the object. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***LIBCRTAUT:** The system determines the authority for the object by using the value specified on the CRTAUT parameter on the Create Library command (CRTLIB) for the library containing the object to be created. If the value specified on the CRTAUT parameter is changed, the new value will not affect any existing objects.

***CHANGE:** The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority.

***ALL:** The user can control the object's existence, specify the security for the object, change the object, change the owner for the object, and perform basic functions on the object. All authority allows the user to perform all operations on the object except those limited to the owner, or controlled by authorization list management authority.

***USE:** The user can perform basic operations on the line description, such as running a program or reading a file. The user cannot change the line description. *USE authority provides object operational authority and read authority.

***EXCLUDE:** The user cannot access the line description.

authorization-list-name: Specify the name of an authorization list. Users included on the authorization list are granted authority to the object as specified by the list. The authorization list must exist when the object is created.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

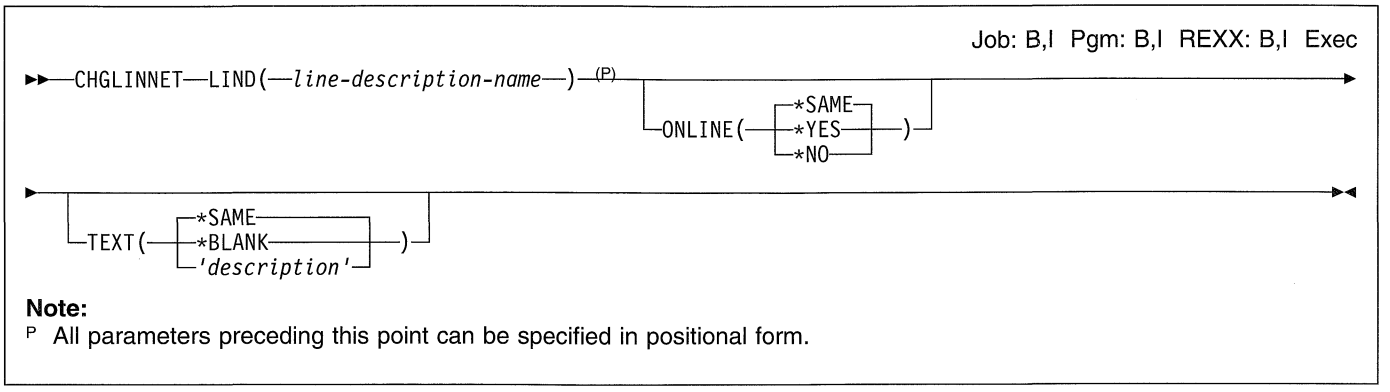
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGLINIDLC   LIND(IDLCLINE)  ONLINE(*NO)
```

This command changes the online parameter to *NO. description named IDLCLINE to LIN011

CHGLINNET (Change Line Description (Network)) Command



Note:

| P All parameters preceding this point can be specified in positional form.

Purpose

The Change Line Description (Network) (CHGLINNET) command changes a line description for a network line. More information about using this command is in the *OS/400* Communications Configuration Reference* manual.

Required Parameter

LIND

Specifies the name of the line description being changed.

Optional Parameters

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** This value does not change.

***YES:** The line is automatically varied on at IPL.

***NO:** This line is not automatically varied on at IPL.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

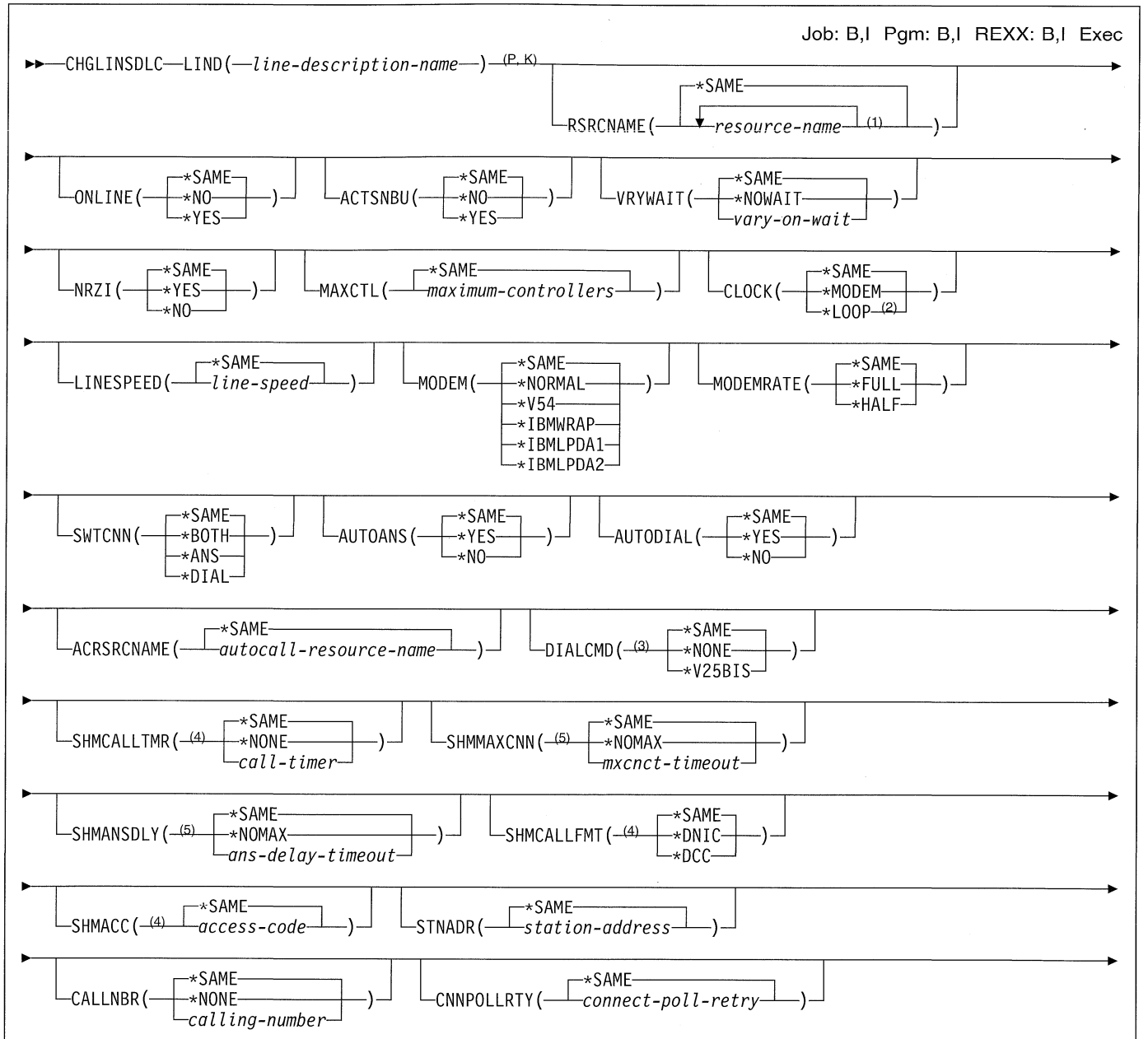
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGLINNET  LIND(LIN0A)  ONLINE(*NO)
```

This command changes the ONLINE parameter value to *NO for a network line named LIN0A.

CHGLINSDLC (Change Line Description (SDLC)) Command



Notes:

1 A maximum of 6 repetitions

2 This value is valid only if the Input/Output Adapter (IOA) port uses the RS-449/V.36 physical interface.

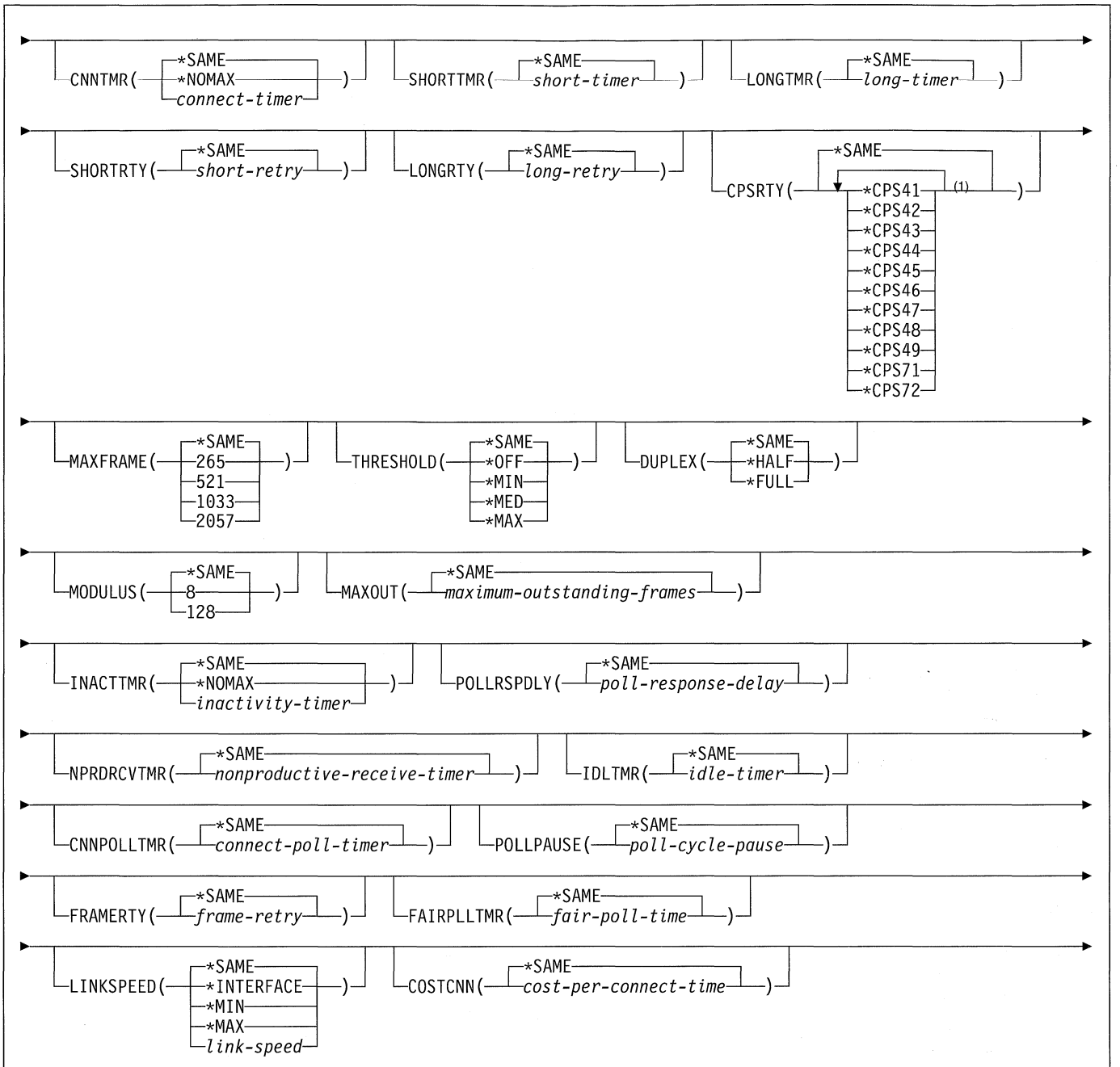
3 This parameter applies only if *YES is specified on the AUTODIAL parameter.

4 This parameter is valid only if *SHM is specified on the CNN parameter.

5 This parameter is valid only if *PRI or *NEG is specified on the ROLE parameter and *SHM is specified on the CNN parameter.

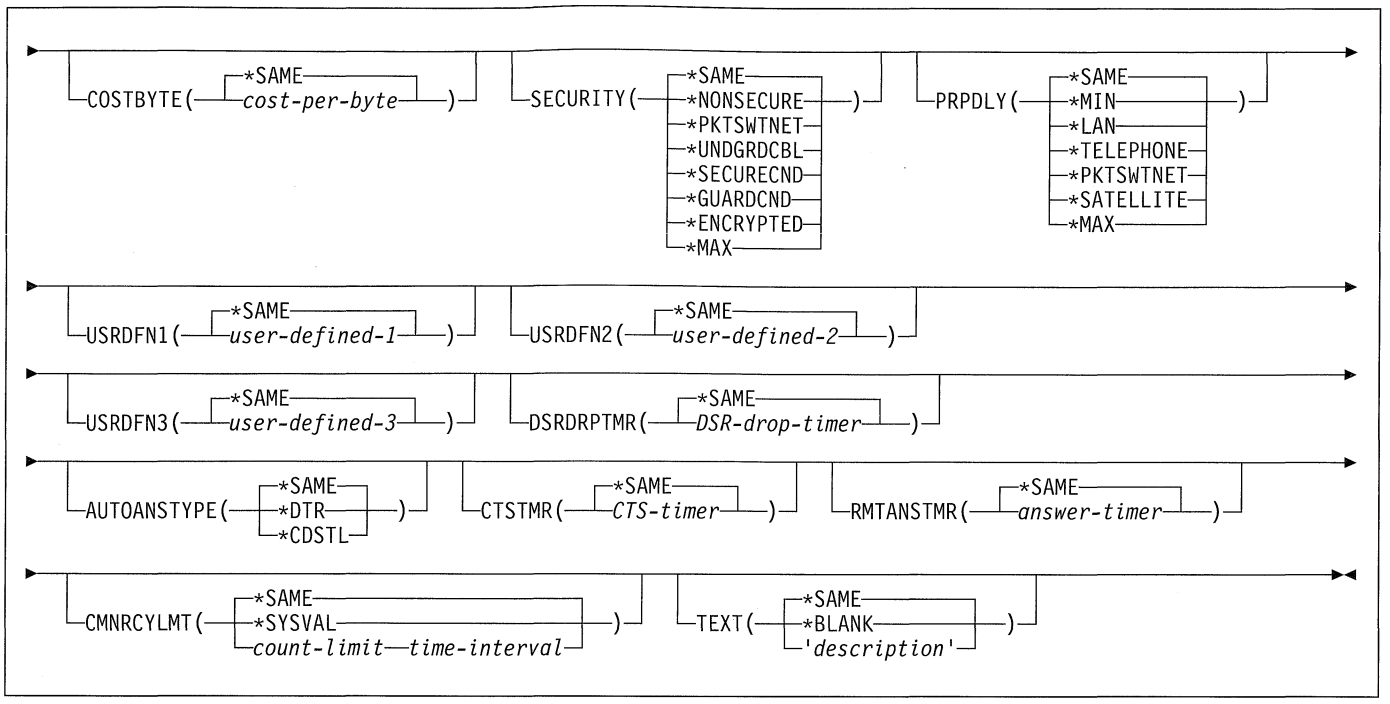
P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.



Note:

[1] A maximum of 11 repetitions



Purpose

The Change Line Description (SDLC) (CHGLINSDLC) command changes a line description for a synchronous data link control (SDLC) line.

Required Parameter

LIND

Specifies the name of the line description being changed.

Optional Parameters

RSRCNAME

Specifies the resource name that describes the automatic call unit port.

Note: Use the Work with Hardware Resources (WRKHDWRSC) command with *CMN specified for the TYPE parameter to help determine the resource name.

***SAME:** The value does not change.

resource-name: Specify the resource name of the communications port. The resource name consists of the input/output adapter (IOA) resource name and the port number on the IOA. For example, if the resource name of the IOA is LIN01 and the port on the IOA is 2, then the resource name would be LIN012.

Up to six resource names can be specified if CNN(*SHM) is specified (X.21 short hold mode). However, if CNN(*SHM) is not specified, or if CNN(*SHM), ROLE(*SEC), and NODETYPE(*T20) are specified, only one resource name can be used.

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***NO:** This line is not automatically varied on at IPL.

***YES:** The line is automatically varied on at IPL.

ACTSNBU

Specifies, for controllers supporting the switched network backup (SNBU) feature, whether the SNBU feature is activated or deactivated. Both the local and remote modems must support the SNBU feature to perform a valid activation.

| ***SAME:** The value does not change from the value previously specified either on this parameter or on the SNBU parameter of the Create Line Description (SDLC) (CRTLINSDLC) command.

***NO:** The switched network backup (SNBU) feature is not activated.

***YES:** The switched network backup feature (SNBU) is activated. To start the feature when the nonswitched connection is broken, specify ACTSNBU(*YES) on the line description command.

VRYWAIT

Specifies whether the line is varied on asynchronously or synchronously. For synchronous vary on, specifies how long the system waits for the vary on to complete.

***SAME:** The value does not change.

| ***NOWAIT:** The system does not wait for the vary on to complete. The line is varied on asynchronously.

| *vary-on-wait:* Specify the time (in seconds) to wait. Valid values range from 15 through 180. The system

waits until the line is varied on, or until the specified time passes, before completing the Vary Configuration (VRYCFG) command.

Notes:

1. Specifying a wait time in the line description affects system IPL time, if ONLINE(*YES) is used, by the amount of time it takes to synchronously vary on the line or reach the wait-time value.
2. The time required to vary on a line is the time it takes to put tasks in place to manage the line, to activate the communications I/O processor (IOP) (including downloading the IOP model-unique Licensed Internal Code), and to establish communications with the data circuit-terminating equipment (DCE). Normal vary-on time ranges from 5 through 45 seconds, but can be longer, depending on the system, line protocol, and other factors.

NRZI

Specifies whether non-return-to-zero-inverted (NRZI) data encoding is used for modems that are sensitive to certain bit patterns in the data stream. This ensures that the signal does not stay the same for an extended period of time.

Note: All data communications equipment on the line must use the same transmission method.

***SAME:** The value does not change.

***YES:** NRZI data encoding is used.

***NO:** NRZI data encoding is not used.

MAXCTL

Specifies the maximum number of controllers that the line supports.

***SAME:** The value does not change.

maximum-controllers: Specify the maximum number of controllers that the line supports. Valid values range from 1 through 254. Specify a number large enough to account for all of the controllers that are currently active to this line, and for any controllers to be attached in the near future.

CLOCK

Specifies how the clocking function for the line is provided.

***SAME:** The value does not change.

***MODEM:** The modem supplies the clocking function.

***LOOP:** The receiving clock provided by the modem data circuit-terminating equipment (DCE) is looped back to the modem DCE on the system data terminal equipment (DTE) transmitting clock. This option can be used to improve high speed data transmission when the modem DCE supports such an option.

LINESPEED

Specifies the line speed in bits per second (bps).

***SAME:** The value does not change.

line-speed: Specify one of the following line speeds: 600, 1200, 2400, 4800, 7200, 9600, 14400, 19200, 48000, 56000, 57600, 64000, 112000, 128000, 168000, 192000, 224000, 256000, 280000, 320000, 336000, 384000, 392000, 448000, 504000, 512000, 560000, 576000, 616000, 640000, 672000, 704000, 728000, 768000, 784000, 832000, 840000, 896000, 952000, 960000, 1008000, 1024000, 1064000, 1088000, 1120000, 1152000, 1176000, 1216000, 1232000, 1280000, 1288000, 1344000, 1400000, 1408000, 1456000, 1472000, 1512000, 1536000, 1568000, 1600000, 1624000, 1664000, 1680000, 1728000, 1736000, 1792000, 1856000, 1920000, 1984000, or 2048000 bits per second.

MODEM

Specifies the type of modem supported on the communications line. Refer to the user's modem documentation to select the appropriate value.

***SAME:** The value does not change.

***NORMAL:** No attempt is made to run diagnostic tests on the modem.

***V54:** A certain type of diagnostics (as defined by the CCITT recommendations) are run to the modem. The AS/400 system supports CCITT V.54 loop 3, local loop back, and loop 2, which is a remote loop back.

***IBMWRAP:** An IBM modem with wrap test capabilities is used on the communications line.

***IBMLPDA1:** An IBM modem with Link Problem Determination Aid-1 (LPDA-1) is used on this line.

***IBMLPDA2:** An IBM modem with Link Problem Determination Aid-2 (LPDA-2) is used on this line.

MODEMRATE

Specifies the speed at which the line operates if the modem has the data rate select feature.

***SAME:** The value does not change.

***FULL:** The line operates at the full rate of the modem.

***HALF:** The line operates at half the full rate, or at the alternate rate of the modem.

SWTCNN

Specifies whether the switched or switched network backup line is used for incoming calls, outgoing calls, or both.

***SAME:** The value does not change.

***BOTH:** The line is used for both incoming and outgoing calls.

***ANS:** The line is used for incoming calls only.

***DIAL:** The line is used for outgoing calls only.

AUTOANS

Specifies, for switched or switched network backup (SNBU) lines, whether the system automatically answers a call from a remote system to establish the connection

or whether the user must manually answer the call and place the modem in data mode.

Note: Interface automatic answering must be specified for X.21 circuit switched lines

***SAME:** The value does not change.

***YES:** The AS/400 system automatically answers a call.

***NO:** The system operator must manually answer a call.

Note: *YES is valid only if the modem has the automatic answer feature or if an X.21 circuit switched interface is used.

AUTODIAL

Specifies, for switched or switched network backup (SNBU) lines, whether the system automatically calls a remote system to establish a connection or whether the system operator must manually place the call.

Note: For X.21 circuit switched lines, automatic dialing must be specified.

***SAME:** The value does not change.

***YES:** The AS/400 system automatically calls a remote system.

***NO:** The system operator must manually call a remote system.

Note: *YES is valid only if the system is using an autocall unit, X.21 circuit switched interface, or if the modem is capable of calling through a command interface such as V.25 bis.

ACRSRCNAME

Specifies the resource name that describes the automatic call unit port.

***SAME:** The value does not change.

autocall-resource-name: Specify the resource name of the autocall unit port. The resource name consists of the input/output adapter (IOA) resource name and the port number on the IOA. For example, if the resource name of the IOA is LIN02 and the port is 1, then the resource name would be LIN021.

DIALCMD

Specifies the type of dial command used to establish a connection with a remote system.

***SAME:** The value does not change.

***NONE:** No dial command type is specified. (Autocall unit or X.21 circuit switched interface.)

***V25BIS:** V.25 bis is a recommendation that uses one physical interface for call establishment and data transmission. It is referred to as a serial automatic call interface because the digits are presented serially on the link from the system (DTE) to the modem (DCE).

SHMCALLTMR

Specifies the interval at which a connection is reestablished on an X.21 short hold mode line to verify the state of the remote system even if no normal data traffic has occurred in the specified interval.

Note: This parameter is valid only if CNN(*SHM) is specified.

***SAME:** The value does not change.

***NONE:** No call is made to verify the connection.

call-timer: Specify an interval, ranging from 1 through 60 minutes, at which a call is made to verify the connection.

SHMMAXCNN

Specifies the amount of time the system allows a connection to continue if there are more controllers than there are available ports. The system clears the connection after the specified amount of time, delay further calls for the amount of time specified on the SHMANSDLY parameter, and then make any calls that have been waiting before re-calling the controller that was interrupted.

Note: This parameter is valid only if *PRI or *NEG is specified on the ROLE parameter and *SHM is specified on the CNN parameter.

***SAME:** The value does not change.

***NOMAX:** The timer is disabled.

mxcnct-timeout: Specify a value ranging from 1 through 254 seconds.

SHMANSDLY

Specifies the amount of time the system waits for controllers to call in before making outgoing calls.

Note: This parameter is valid only if *PRI or *NEG is specified on the ROLE parameter and *SHM is specified on the CNN parameter.

***SAME:** The value does not change.

***NOMAX:** The timer is disabled.

ans-delay-timeout: Specify a value ranging from 1 through 254 tenths of a second. For example, 10 seconds equals 100 tenths of a second.

SHMCALLFMT

Specifies the format for the X.21 short hold mode line access code. This parameter is valid only if CNN(*SHM) is specified.

***SAME:** The value does not change.

***DNIC:** A four-digit Data Network Identification Code is used for the short hold mode access code (SHMACCCODE parameter). This code is attached in front of the calling number specified by the CALLNBR parameter.

***DCC:** A three-digit Data Country Code is used for the short hold mode access code (SHMACCCODE parameter). This code is attached in front of the calling number specified by the CALLNBR parameter.

SHMACC

Specifies the access code used by an X.21 short hold mode line when calling a system on another network. The access code can be three or four digits, depending on the value specified for the SHMCALLFMT parameter.

Note: This parameter is valid only if CNN(*SHM) is specified.

***SAME:** The value does not change.

access-code: Specify the access code.

STNADR

For a switched secondary or negotiable line, this specifies the hexadecimal station address which the local system responds to when polled by the remote system if it answers a call.

***SAME:** The value does not change.

station-address: Specify the local station address using hexadecimal characters ranging from 01 through FE.

CALLNBR

Specifies the local telephone number of the line used for the V.25 bis call request with identification (CRI) dial command. This parameter is used when the CRI function is needed for V.25 bis. When V.25 bis CRI dialing is used, the system takes the called (connection) number from the CNNNBR parameter of the controller description, adds a separator character (;), and concatenates the calling number at the end. Specify the calling number only if the modem and the network both support the CRI dial command.

***SAME:** The value does not change.

***NONE:** The CRN (Call Request Normal) dial command is used by the V.25 bis line.

calling-number: Specify up to 22 characters that represent the local telephone number for V.25 bis CRI (Call Request with Identification) automatic dialing.

This parameter is required for all X.21 short hold mode lines. The calling number is passed to the remote system at the initial connection. This number is dialed by the remote system to reestablish the short hold mode connection.

CNNPOLLRTY

Specifies, for a switched primary or negotiable line, the number of connect poll retries before indicating the error and making the station inoperative.

***SAME:** The value does not change.

connect-poll-retry: Specify a value ranging from 0 through 64 for the number of retries.

CNNTMR

Specifies, for X.21 circuit switched connections, the amount of time an automatic answer connect request waits for an incoming call to be accepted.

***SAME:** The value does not change.

***NOMAX:** There is no disconnect limit.

connect-timer: Specify a value ranging from 1 through 32767 in 0.1 second intervals.

SHORTTMR

Specifies, for X.21 circuit switched connections, the time the system waits between connection attempts. This timer is used during bursts of retry operations.

***SAME:** The value does not change.

short-timer: Specify a value ranging from 10 through 600 in 0.1 second intervals.

LONGTMR

Specifies, for X.21 circuit switched connections, the timer used between bursts of retry operations. After a burst of retry attempts, the system waits for this timeout period before the next attempt.

***SAME:** The value does not change.

long-timer: Specify a value ranging from 100 through 6000 in 0.1 second intervals.

SHORTRTY

Specifies, for X.21 circuit switched connections, the number of retry attempts that are made during a burst of retries.

***SAME:** The value does not change.

short-retry: Specify a value ranging from 0 through 254 for the number of retries.

LONGRTY

Specifies, for X.21 circuit switched connections, the number of burst retry attempts that are made when processing a connect request.

***SAME:** The value does not change.

long-retry: Specify a value ranging from 0 through 254 for the number of retries.

CPSRTY

Specifies whether call progress signals are retried for X.21 circuit switched or X.21 short hold mode lines. Up to 11 values can be specified; duplicate values are ignored.

***SAME** This value does not change.

Valid values are:

*CPS41
*CPS42
*CPS43
*CPS44
*CPS45
*CPS46
*CPS47
*CPS48
*CPS49
*CPS71
*CPS72

This parameter can be specified only if CNN(*SHM) and INTERFACE(*X21) are specified. The *OS/400* Commu-

nications Configuration Reference has descriptions of each of the call progress signals.

MAXFRAME

Specifies the maximum frame (path information unit (PIU)) size that the controller can send or receive. This value is used to calculate request unit (RU) sizes. Since the maximum PIU size that the controller can send or receive is negotiated at exchange identifier time, the maximum PIU size used at run time may be different. This value matches the corresponding value on the host system.

***SAME:** The value does not change.

maximum-frame-size: Specify the maximum frame size. Valid values are:

265
521
1033
2057

THRESHOLD

Specifies the temporary error threshold level being monitored by the system. A permanent error is reported only if the errors occurred consecutively and exceeded the retry limit.

Note: Specifying the THRESHOLD parameter affects all threshold errors. They cannot be specified individually.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** The error threshold is set at a minimum monitoring level.

***MED:** Error thresholding is set to a medium monitoring level.

***MAX:** The error threshold is set at a maximum monitoring level.

DUPLEX

Specifies whether request-to-send (RTS) is permanently turned on (for full-duplex modems) or turned on only when data transmission is required (for half-duplex modems).

Note: For X.21 circuit-switched interface lines, full duplex permanent RTS must be specified.

***SAME:** The value does not change.

***HALF:** RTS turned on only when transmission is required (for half-duplex modems).

***FULL:** Request-to-send (RTS) is permanently turned on (for full-duplex modems).

MODULUS

Specifies whether extended sequence numbers are used.

***SAME:** The value does not change.

8: Extended sequence numbers are not used (modulus 8).

128: Extended sequence numbers are used (modulus 128).

MAXOUT

Specifies the maximum number of frames that are transmitted sequentially to a remote system before the remote system must respond.

For modulus 8, the maximum number of frames ranges from 1 through 7. For modulus 128, the maximum number of frames ranges from 8 through 28.

***SAME:** The value does not change.

maximum-outstanding-frames: Specify a value ranging from 1 through 28 for the number of frames.

INACTTMR

Specifies, for a secondary or negotiable line, the time (in tenths of a second) that the system waits for a valid frame to flow before reporting the error and disconnecting the line. This timer is started at connection time, restarted upon transmission of any frame, and reset upon receipt of a frame with a valid frame check sequence.

***SAME:** The value does not change.

***NOMAX:** There is no disconnect limit.

inactivity-timer: Specify a value ranging from 150 through 4200 in 0.1 second intervals.

POLLRSPDLY

Specifies, for a secondary or negotiable line, the minimum time that the system waits before it responds to a data poll if there is no frame to transmit.

***SAME:** The value does not change.

poll-response-delay: Specify a value ranging from 1 through 2048 in 0.0001 second intervals, or specify 0 to indicate no delay.

NPRDRCVTMR

Specifies, for a primary or negotiable line, the time that the system waits for either a final frame or an idle signal while the secondary station is continuously transmitting. If this timer expires, then the nonproductive receive condition is reported.

***SAME:** The value does not change.

nonproductive-receive-timer: Specify a value ranging from 160 through 4200 in 0.1 second intervals.

IDLTMR

Specifies, for a primary or negotiable line, the time (in 0.1 second intervals) that the system waits before sampling the line for an idle signal. If an idle signal is found, error recovery procedures are started.

***SAME:** The value does not change.

idle-timer: Specify a value ranging from 5 through 300 in 0.1 second intervals.

CNNPOLLTMR
Specifies, for a primary or negotiable line, the time that the system waits for the response to a connect poll before retransmitting the poll.

***SAME:** The value does not change.

connect-poll-timer: Specify a value ranging from 2 through 300 in 0.1 second intervals.

POLLPAUSE
Specifies, for a primary or negotiable line, the time that the system pauses after the last remote system in the poll list is polled.

***SAME:** The value does not change.

poll-cycle-pause: Specify a value ranging from 1 through 2048 in 0.0001 second intervals, or specify 0 to indicate no pause.

FRAMERTY
Specifies the number of retries for an unanswered command frame or unacknowledged information frame before indicating the error and disconnecting the station.

***SAME:** The value does not change.

frame-retry: Specify a value ranging from 0 to 64 for the number of retries.

FAIRPLLTMR
Specifies the maximum amount of time for which the system sends data to one or more work stations before requesting input from the work stations.

***SAME:** The value does not change.

fair-poll-time: Specify the amount of time (in seconds) the system waits (during output transfers) before polling for input. Valid values for this parameter range from 5 through 60 seconds.

LINKSPEED
Specifies the link speed in bits per second (bps). This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

***INTERFACE:** The following link speeds, based on the physical interface type, are used: 9600 bps for RS-232/V.24 and X.21bis/V.24, 48000 bps for V.35 and X.21bis/V.35, and 64000 bps for X.21 and RS-449/V.36.

***MIN:** A link speed of less than 1200 bps is used.

***MAX:** A link speed greater than 16M bps is used.

link-speed: Specify the link speed. Valid values are: 1200, 2400, 4800, 7200, 9600, 14400, 19200, 48000, 56000, 64000, 112000, 128000, 168000, 192000, 224000, 256000, 280000, 320000, 336000, 384000, 448000, 499000, 576000, 614000, 691000, 768000, 845000, 922000, 998000, 1075000, 1152000, 1229000, 1382000, 1536000, 1690000, 1843000, 1997000, 2048000, 4M, 10M, and 16M.

COSTCNN
Specifies the relative cost of being connected on the line. This parameter is required only if APPN is used on the system.

***SAME:** The value does not change.

cost-per-connect-time: Specify a value ranging from 0 through 255.

COSTBYTE
Specifies the relative cost per byte for sending and receiving data on the line. This parameter is required only if APPN is used on the system.

***SAME:** The value does not change.

cost-per-byte: Specify a value ranging from 0 through 255.

SECURITY
Specifies the security level of the physical line. This parameter is used only if APPN is used on the system.

***SAME:** The value does not change.

***NONSECURE:** Normal priority is used.

***PKTSWTNET:** A packet switched network is used. Data does not always follow the same path through the network.

***UNDGRDCBL:** An underground cable is used.

***SECURECND:** A secure, unguarded conduit (for example, a pressurized pipe) is used.

***GUARDCN:** A guarded conduit, which is protected against physical tapping, is used.

***ENCRYPTED:** Data flowing on the line is encrypted.

***MAX:** A guarded conduit, protected against physical and radiation tapping, is used.

PRPDLY
Specifies the level of propagation delay on the line. This parameter is valid only if APPN is used on the system. The order of the values from shortest to longest delay is *MIN, *LAN, *TELEPHONE, *PKTSWTNET, and *SATELLITE.

***SAME:** The value does not change.

***MIN:** The minimum propagation delay is used.

***LAN:** The local area network propagation delay is used.

***TELEPHONE:** The telephone propagation delay is used.

***PKTSWTNET:** The packet switched network propagation delay is used.

***SATELLITE:** The satellite propagation delay is used.

***MAX:** The maximum propagation delay is used.

USRDFN1
Specifies the first of the three user-defined fields. This field is used to describe unique characteristics of the line

that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

user-defined-1: Specify a value ranging from 0 through 255.

USRDFN2

Specifies the second of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

user-defined-2: Specify a value ranging from 0 through 255.

USRDFN3

Specifies the third of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

user-defined-3: Specify a value ranging from 0 through 255.

DSRDRPTMR

Specifies the amount of time the system waits for the modem to exit the Data Set Ready (DSR) state before signaling an error.

***SAME:** The value does not change.

DSR-drop-timer: Specify a value ranging from 3 through 60 seconds.

AUTOANSTYP

Specifies the method the system uses to answer incoming calls.

***SAME:** The value does not change.

***DTR:** The system enters the Data Terminal Ready state, signals the modem to answer calls, and waits for the modem to enter the Data Set Ready (DSR) state.

***CDSTL:** The system enters the Connect Data Set to Line (CDSTL) state after monitoring the Ring Indicator to signal the modem to answer the call.

CTSTMR

Specifies the amount of time the system waits for the modem to enter or exit the Clear to Send (CTS) state before signaling an error.

***SAME:** The value does not change.

CTS-timer: Specify a value ranging from 10 through 60 seconds.

RMTANSTMR

Specifies the amount of time the system waits for the modem to enter the Data Set Ready (DSR) state after dialing before signaling an error.

***SAME:** The value does not change.

answer-timer: Specify a value ranging from 30 through 120 seconds.

CMNRCYLMT

Specifies the number of recovery attempts made by the system before an inquiry message is sent to the system operator. Also specifies the time (in minutes) that must elapse before the system sends an inquiry message to the system operator indicating that the recovery attempt count limit is reached.

***SAME:** The value does not change.

***SYSVAL:** The recovery limits specified in the QCMNRCYLMT system value are used.

Element 1: Maximum Recovery Limit

count-limit: Specify the number of recovery attempts to be made. Valid values range from 0 through 99.

Element 2: Recovery Time Interval

time-interval: Specify the time interval (in minutes) at which the specified number of second-level recoveries are attempted. Valid values range from 0 through 120. If the value specified for *count-limit* is not 0, the value 0 specifies infinite recovery.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

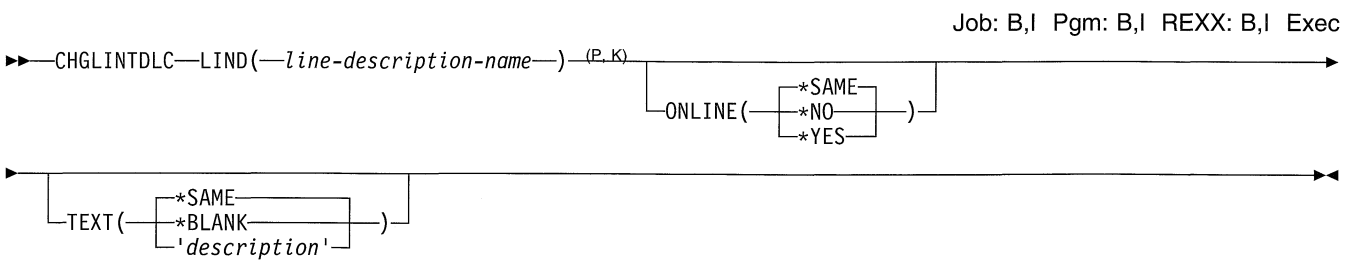
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGLINSDLC   LIND(TUESDAY)  RSRCNAME(LIN031)
```

This command changes the resource name for line description TUESDAY to LIN031.

CHGLINTDLC (Change Line Description (TDLC)) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Line Description (TDLC) (CHGLINTDLC) command changes a line description for a Twinaxial Data Link Control (TDLC) line.

Required Parameter

LIND

Specifies the name of the line description being changed.

Optional Parameters

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***NO:** This line is not automatically varied on at IPL.

***YES:** The line is automatically varied on at IPL.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

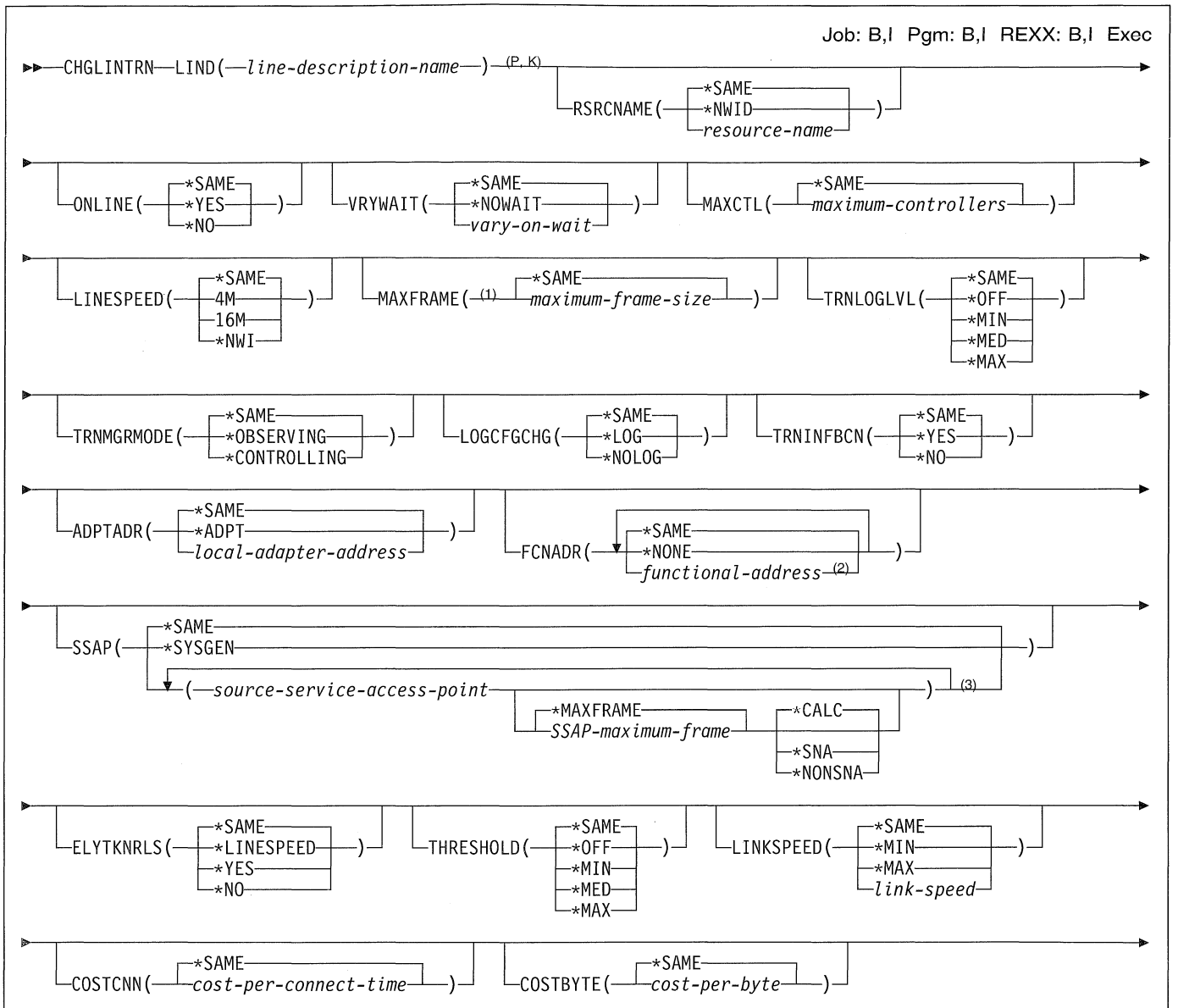
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGLINTDLC  LIND(WSFLIN1)  TEXT('Virtual line for WSF')
```

This command changes the text for line description WSFLIN1.

CHGLINTRN (Change Line Description (Token-Ring Network)) Command



Notes:

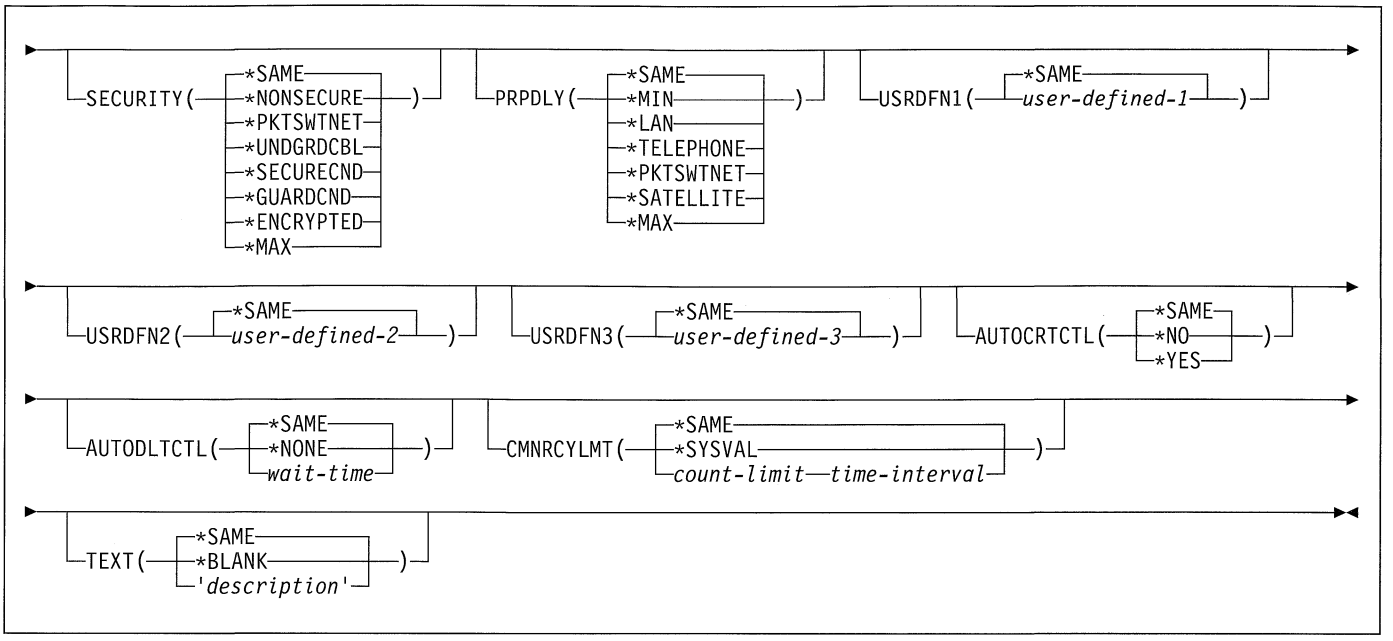
P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

1 When RSRCNAME(*NWID) is specified, any value ranging from 265 through 8151 can be specified. Otherwise, the valid values range from 256 through 16393.

2 A maximum of 31 repetitions

3 A maximum of 24 repetitions



Purpose

The Change Line Description (Token-Ring Network) (CHGLINTRN) command changes a line description for a Token-Ring Network line.

Required Parameter

LIND

Specifies the name of the line description being changed.

Optional Parameters

RSRCNAME

Specifies the resource name that describes the automatic call unit port.

Note: Use the Work with Hardware Resources (WRKHDWSRC) command with *CMN specified for the TYPE parameter to help determine the resource name.

The value specified on the RSRCNAME parameter cannot be changed from *NWID to another value or from another value to *NWID.

***SAME:** The value does not change.

***NWID:** The resource name is determined by the network interface used.

resource-name: Specify the resource name of the communications port. The resource name consists of the input/output adapter (IOA) resource name and the port number on the IOA. For example, if the resource name of the IOA is LIN01 and the port on the IOA is 1, then the resource name would be LIN011.

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The line is automatically varied on at IPL.

***NO:** This line is not automatically varied on at IPL.

VRYWAIT

Specifies whether the line is varied on asynchronously or synchronously. For synchronous vary on, specifies how long the system waits for the vary on to complete.

***SAME:** The value does not change.

***NOWAIT:** The system does not wait for the vary on to complete. The line is varied on asynchronously.

vary-on-wait: Specify the time (in seconds) to wait. Valid values range from 15 through 180. The system waits until the line is varied on, or until the specified time passes, before completing the Vary Configuration (VRYCFG) command.

Notes:

1. Specifying a wait time in the line description affects system IPL time, if ONLINE(*YES) is used, by the amount of time it takes to synchronously vary on the line or reach the wait-time value.
2. The time required to vary on a line is the time it takes to put tasks in place to manage the line, to activate the communications I/O processor (IOP) (including downloading the IOP model-unique Licensed Internal Code), and to establish communications with the data circuit-terminating equipment (DCE). Normal vary-on time ranges from 5 through 45 seconds, but can be longer, depending on the system, line protocol, and other factors.

MAXCTL

Specifies the maximum number of controllers that the line supports.

***SAME:** The value does not change.

maximum-controllers: Specify the maximum number of controllers the line will support. Valid values range from 1 through 256. Specify a number large enough to account for all of the controllers currently active to this network, and allow for controllers that may be attached in the near future.

LINESPEED

Specifies the line speed in bits per second (bps).

Note: When RSRCNAME(*NWID) is specified, *NWI must be specified on this parameter.

***SAME:** The value does not change.

4M: The line speed is 4M bits per second (bps).

16M: The line speed is 16M bits per second (bps).

***NWI:** The linespeed is used from the network interface. This value is valid only when RSRCNAME(*NWID) is specified.

MAXFRAME

Specifies the maximum frame size that can be transmitted and received on this line description.

***SAME:** The value does not change.

maximum-frame-size: Specify the maximum frame size. When RSRCNAME(*NWID) is specified, valid values range from 265 through 8148 bytes. Otherwise, valid values range from 265 through 16393 bytes. Suggested values for this parameter include, but are not limited to, the following:

```
265   1033   1600   4060    16393
521   1466   1994   8156
```

TRNLOGLVL

Specifies the error logging level used by the IBM Token-Ring Network (TRLAN) Manager.

Note: TRNLOGLVL(*OFF) must be specified when RSRCNAME(*NWID) is specified.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** The minimum error logging level, which reports only conditions that indicate degraded performance, is used.

***MED:** The medium error logging level, which reports conditions that indicate potential degraded performance, is used.

***MAX:** The maximum error logging level, which reports all error conditions, including those reported for the *MIN and *MED reporting levels, is used.

TRNMGRMODE

Specifies whether the token-ring network manager is controlling or observing.

Note: TRNMGRMODE(*OBSERVING) must be specified when RSRCNAME(*NWID) is specified.

***SAME:** The value does not change.

***OBSERVING:** The token-ring manager is observing.

***CONTROLLING:** The token-ring manager is controlling.

LOGCFGCHG

Specifies whether the nearest active upstream neighbor (NAUN) changes are logged.

Note: LOGCFGCHG(*LOG) must be specified when RSRCNAME(*NWID) is specified.

***SAME:** The value does not change.

***LOG:** NAUN changes are logged.

***NOLOG:** NAUN changes are not logged.

TRNINFBCN

Specifies whether an inform message is sent to QSYSOPR when beaconing occurs.

Note: TRNINFBCN(*YES) must be specified when RSRCNAME(*NWID) is specified.

***SAME:** The value does not change.

***YES:** A message is sent to QSYSOPR when beaconing occurs.

***NO:** A message is not sent to QSYSOPR when beaconing occurs.

ADPTADR

Specifies the 12-character hexadecimal adapter address.

Note: ADPTADR(*ADPT) is not valid when RSRCNAME(*NWID) is specified.

***SAME:** The value does not change.

***ADPT:** This value gives the user the preset token-ring default address for this token-ring adapter card. This may be displayed by using the DSPLIND command for this line description after it has successfully varied on.

local-adapter-address: Specify an address to this system in this token-ring network. Valid values range from hexadecimal 400000000000 through 7FFFFFFFFFFF.

FCNADR

Specifies whether token ring functional addresses are used.

***SAME:** The value does not change.

***NONE:** A functional address is not used.

functional-address: Specify a group of hexadecimal functional addresses that are encoded in bit-significant format. Valid values range from hex C00000000001 through hex C00040000000. The first digit must be C. Functional addresses must be unique.

SSAP

Specifies the source service access point (SSAP). The most commonly used SNA SSAP is hex 04. All SSAP values must be unique.

***SAME:** The value does not change.

***SYSGEN:** The system automatically creates three SSAPs, hex 04 for SNA, hex AA and 06 for Transmission Control Protocol/Internet Protocol (TCP/IP) applications.

Element 1: User-defined SSAPs for TCP/IP

source-service-access-point: Specify up to 16 SSAPs, including hex AA and hex 06 for TCP/IP applications, and any hexadecimal number ranging from 04 through 9C that is divisible by 4 for SNA applications.

Element 2: Frame Size for SSAPs

***MAXFRAME:** The frame size specified on the MAXFRAME parameter is used.

SSAP-maximum-frame-size: Specify the maximum SSAP frame size (the maximum size of the data field which may be transmitted or When RSRCNAME(*NWID) is specified, valid values for this parameter range from 265 through 8148. Otherwise, valid values for this parameter range from 265 through 16393. The commonly used values for this parameter are 256, 1033, 1466, 1994, 4060, and 8156.

Note: The values specified on this parameter cannot be larger than the values specified on the MAXFRAME parameter.

Element 3: SSAP Type

***CALC:** The system determines the SSAP type based on the following values:

- 04 through 9C, divisible by 4 (for SNA)
- 02 through FE, divisible by 2 (for non-SNA)

***SNA:** The SSAP is used for SNA communications. Only SSAP values of 04 through 9C and must be divisible by 4 are supported.

***NONSNA:** The SSAP is used for non-SNA Communications. Only SSAP values of 02 through FE and must be divisible by 2 are supported.

ELYTKNRLS

Specifies whether the early token release option is used.

Note: ELYTRNRLS(*LINESPEED) must be specified when RSRCNAME(*NWID) is specified.

***SAME:** The value does not change.

***LINESPEED:** The value specified on the LINESPEED parameter is used to determine whether the early token release option is used.

***YES:** The early token release option is used. *YES cannot be specified if the LINESPEED parameter contains the value 4M.

***NO:** The early token release option is not used.

THRESHOLD

This parameter, and its values *SAME, *OFF, *MIN, *MED, and *MAX, can be specified but it is not used by the system starting in release V2R3M0. The parameter may be removed in a later release.

LINKSPEED

Specifies the link speed in bits per second (bps). This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

***MIN:** A link speed of less than 1200 bps is used.

***MAX:** A link speed greater than 16M bps is used.

link-speed: Specify the link speed. Valid values are: 1200, 2400, 4800, 7200, 9600, 14400, 19200, 48000, 56000, 64000, 112000, 128000, 168000, 192000, 224000, 256000, 280000, 320000, 336000, 384000, 448000, 499000, 576000, 614000, 691000, 768000, 845000, 922000, 998000, 1075000, 1152000, 1229000, 1382000, 1536000, 1690000, 1843000, 1997000, 4M, 10M, and 16M.

COSTCNN

Specifies the relative cost of being connected on the line. This parameter is required only if APPN is used on the system.

***SAME:** The value does not change.

cost-per-connect-time: Specify a value ranging from 0 through 255.

COSTBYTE

Specifies the relative cost per byte for sending and receiving data on the line. This parameter is required only if APPN is used on the system.

***SAME:** The value does not change.

cost-per-byte: Specify a value ranging from 0 through 255.

SECURITY

Specifies the security level of the physical line. This parameter is used only if APPN is used on the system.

***SAME:** The value does not change.

***NONSECURE:** Normal priority is used.

***PKTSWTNET:** A packet switched network is used. Data does not always follow the same path through the network.

***UNDGRDCBL:** An underground cable is used.

***SECURECND:** A secure, unguarded conduit (for example, a pressurized pipe) is used.

***GUARDCND:** A guarded conduit, which is protected against physical tapping, is used.

***ENCRYPTED:** Data flowing on the line is encrypted.

***MAX:** A guarded conduit, protected against physical and radiation tapping, is used.

PRPDLY
Specifies the level of propagation delay on the line. This parameter is valid only if APPN is used on the system. The order of the values from shortest to longest delay is *MIN, *LAN, *TELEPHONE, *PKTSWTNET, and *SATELLITE.

***SAME:** The value does not change.

***MIN:** The minimum propagation delay is used.

***LAN:** The local area network propagation delay is used.

***TELEPHONE:** The telephone propagation delay is used.

***PKTSWTNET:** The packet switched network propagation delay is used.

***SATELLITE:** The satellite propagation delay is used.

***MAX:** The maximum propagation delay is used.

USRDFN1
Specifies the first of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

user-defined-1: Specify a value ranging from 0 through 255.

USRDFN2
Specifies the second of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

user-defined-2: Specify a value ranging from 0 through 255.

USRDFN3
Specifies the third of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

user-defined-3: Specify a value ranging from 0 through 255.

AUTOCRTCTL
Specifies whether the system will automatically create controller descriptions when calls are received from adjacent systems on the local area network (LAN).

***SAME:** The value does not change.

***NO:** The system will not automatically create a controller description when incoming calls are received.

***YES:** The system will automatically create a controller description when incoming calls are received.

AUTODLTCTL
Specifies the number of minutes an automatically created controller can remain in an idle state (switched from varied on to varied on pending) before the controller description and attached device descriptions are varied off and deleted.

***SAME:** The value does not change.

***NONE:** The system will not automatically delete or vary off the automatically configured, idle controller descriptions.

wait-time: Specify the number of minutes to wait before deleting the automatically configured, idle controller descriptions for this line. Valid values range from 1 to 10,000 minutes.

CMNRCYLMT
Specifies the number of recovery attempts made by the system before an inquiry message is sent to the system operator. Also specifies the time (in minutes) that must elapse before the system sends an inquiry message to the system operator indicating that the recovery attempt count limit is reached.

***SAME:** The value does not change.

***SYSVAL:** The recovery limits specified in the QCMNRCYLMT system value are used.

Element 1: Maximum Recovery Limit

count-limit: Specify the number of recovery attempts to be made. Valid values range from 0 through 99.

Element 2: Recovery Time Interval

time-interval: Specify the time interval (in minutes) at which the specified number of second-level recoveries are attempted. Valid values range from 0 through 120. If the value specified for *count-limit* is not 0, the value 0 specifies infinite recovery.

TEXT
Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

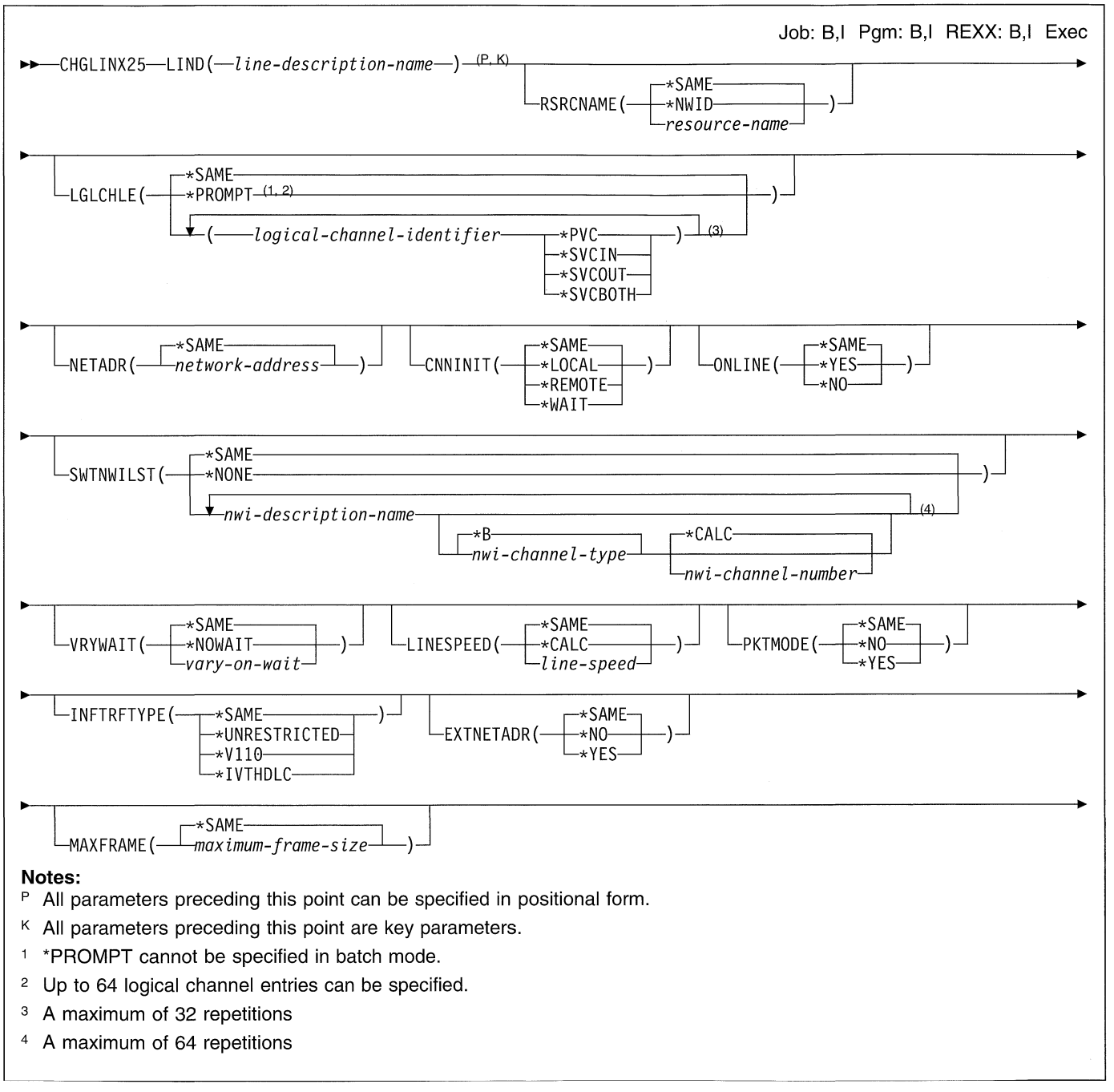
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGLINTRN   LIND(TRLAN1)  MAXCTL(50)
```

This command changes the maximum number of controllers for line description TRLAN1 to 50.

CHGLINX25 (Change Line Description (X.25)) Command



Notes:

P All parameters preceding this point can be specified in positional form.

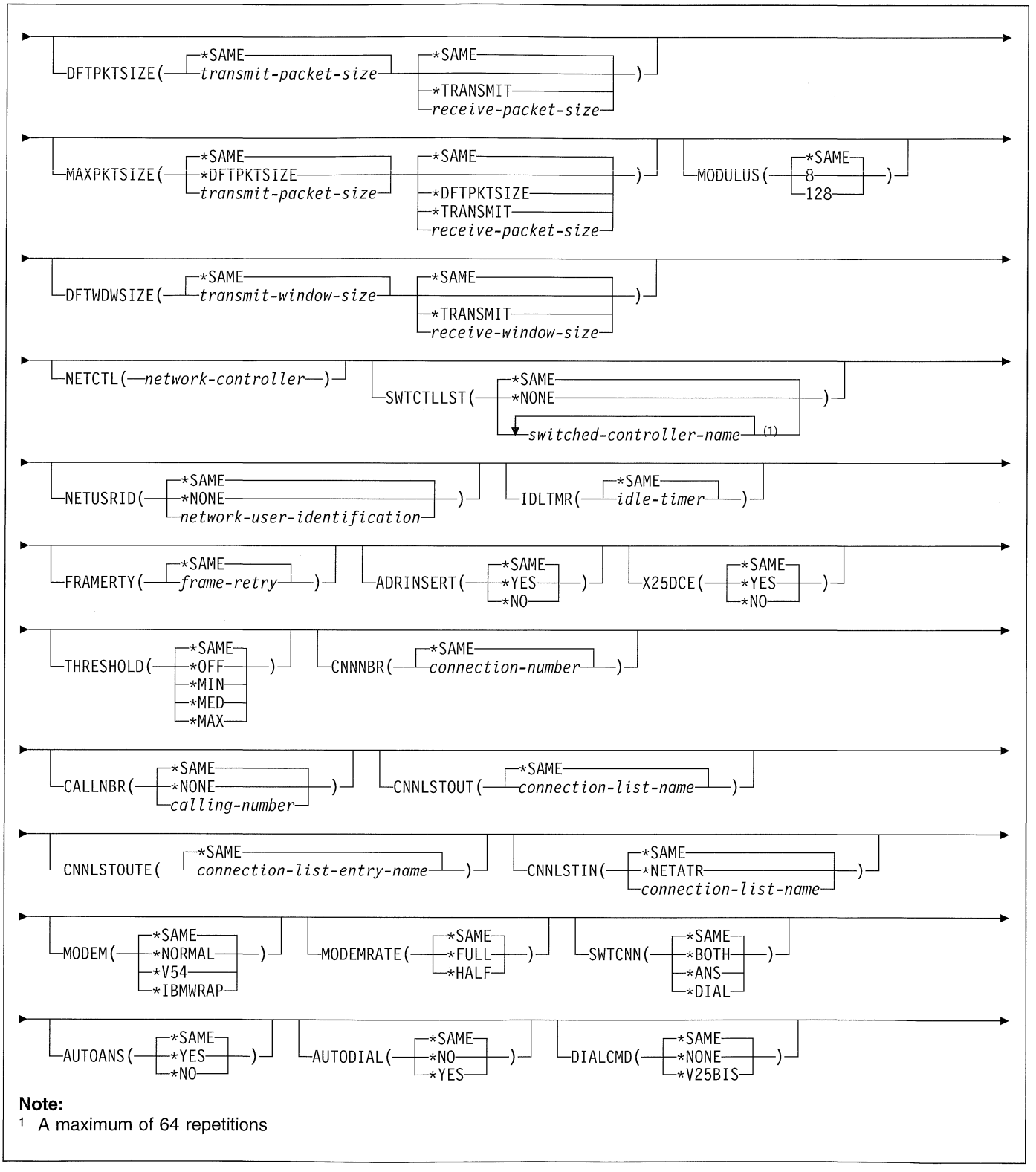
K All parameters preceding this point are key parameters.

1 *PROMPT cannot be specified in batch mode.

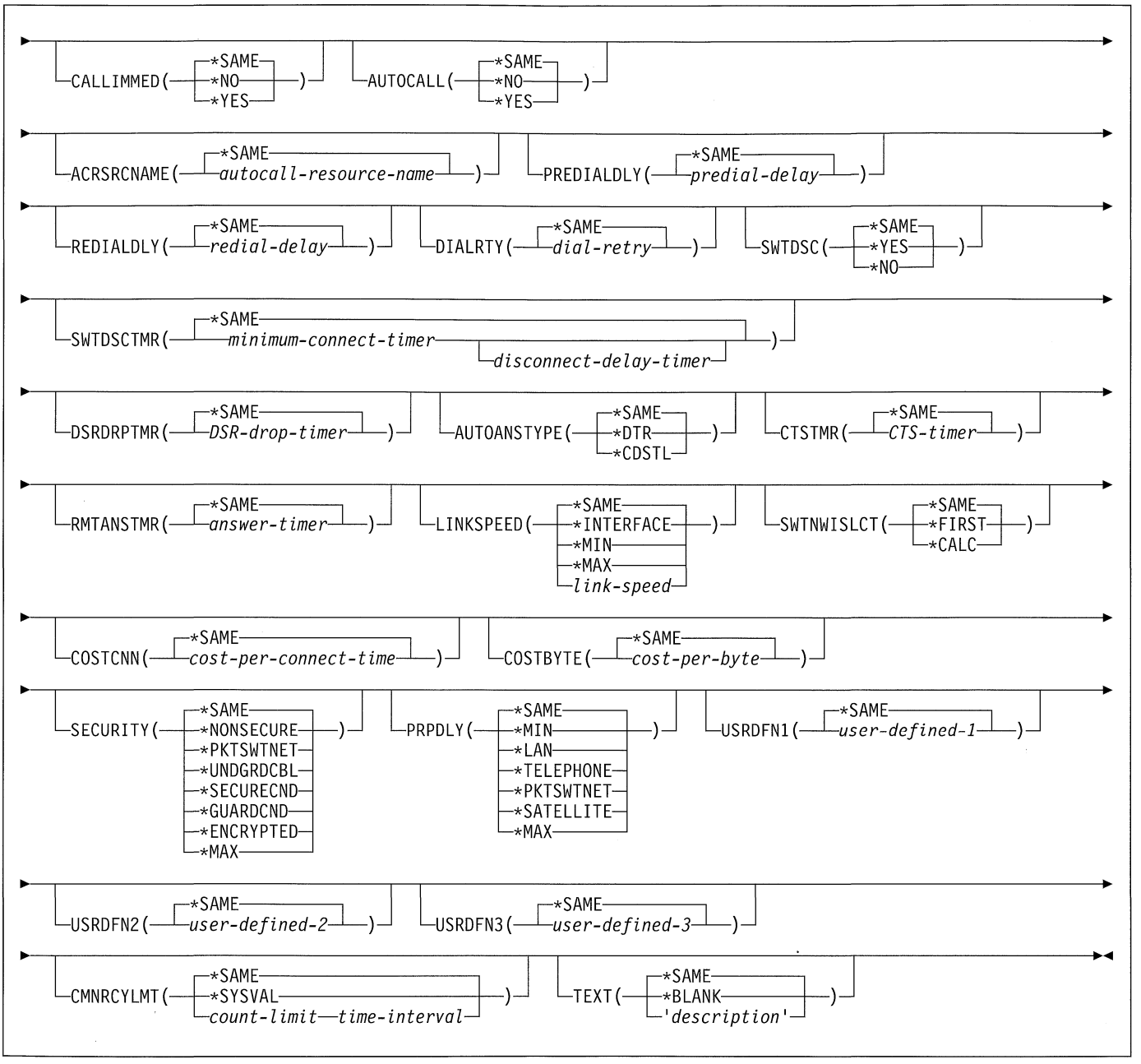
2 Up to 64 logical channel entries can be specified.

3 A maximum of 32 repetitions

4 A maximum of 64 repetitions



Note:
1 A maximum of 64 repetitions



Purpose

The Change Line Description (X.25) (CHGLINX25) command changes a line description for an X.25 line.

Required Parameter

LIND

Specifies the name of the line description being changed.

Optional Parameters

RSRCNAME

Specifies the resource name that describes the automatic call unit port.

Note: Use the Work with Hardware Resources (WRKHDWRSC) command with *CMN specified for the TYPE parameter to help determine the resource name.

The value specified on the RSRCNAME parameter cannot be changed from *NWID to another value or from another value to *NWID.

***SAME:** The value does not change.

***NWID:** The resource name specified on the attached network interface description is used.

resource-name: Specify the resource name of the communications port. The resource name consists of the input/output adapter (IOA) resource name and the port number on the IOA. For example, if the resource name

of the IOA is LIN01 and the port on the IOA is 2, the resource name would be LIN012.

LGLCHLE

Specifies, when using the Create Line Description (X.25) CRTLINX25 or CHGLINX25 command, a list of entries to be added, removed, or changed in the logical channel table. A channel entry consists of a channel identifier, a logical channel type, and a PVC controller.

Note: Logical channel entries with attached permanent virtual circuit (PVC) controllers cannot be added, removed, or changed.

***SAME:** The value does not change.

***PROMPT:** By using the *PROMPT option, any entry can be added, removed, or altered. Specifying *PROMPT shows the current logical channel entries.

Element 1: Logical Channel Identifier

logical-channel-identifier: Specify a value ranging from 001 to FFF for the logical channel identifier. The first digit (from left to right) is the logical channel group number; the second and third digits make up the logical channel number.

Element 2: Logical Channel Type

***PVC:** The logical channel is a permanent virtual circuit.

***SVCIN:** The logical channel is a switched virtual circuit for incoming calls.

***SVCOUT:** The logical channel is a switched virtual circuit for calls going out.

***SVCBOTH:** The logical channel is a switched virtual circuit for both calls coming in and calls going out.

NETADR

Specifies the local network address for this system.

***SAME:** The value does not change.

network-address: Specify up to 15 decimal digits for the network address.

CNNINIT

Specifies the initiator of the X.25 Data Link connection.

***SAME:** The value does not change.

***LOCAL:** The local system initiates the connection (local system sets asynchronous balanced mode (SABM)).

***REMOTE:** The connection is remotely initiated (the local system waits for SABM).

***WAIT:** The local system waits for a disconnect (DISC) or disconnect mode (DM) from the data communication equipment (DCE) before attempting to activate the link.

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The line is automatically varied on at IPL.

***NO:** This line is not automatically varied on at IPL.

SWTNWILST

Specifies, for ISDN switched connections, a list of network interface descriptions to which this line can be attached. A network interface description is chosen from the list based on the value specified on the switched NWI selection parameter (SWTNWISLCT) at the time an incoming or outgoing call is processed.

***SAME:** The value does not change.

***NONE:** No network interface description is specified.

Element 1: Network Interface Description Name

nwi-description-name: Specify, for switched connections, the name of the network interface description to which this line attaches.

Element 2: Network Interface Channel Type

***B:** The B channel is used.

Element 3: Network Interface Channel-Number

***CALC:** The system selects one of the two channel numbers (based on availability) defined for the network interface description when an incoming or outgoing call is processed.

nwi-channel-number: Specify a channel number (1 or 2) to which the line description is restricted.

VRYWAIT

Specifies whether the line is varied on asynchronously or synchronously. For synchronous vary on, specifies how long the system waits for the vary on to complete.

***SAME:** The value does not change.

***NOWAIT:** The system does not wait for the vary on to complete. The line is varied on asynchronously.

vary-on-wait: Specify the time (in seconds) to wait. Valid values range from 15 through 180. The system waits until the line is varied on, or until the specified time passes, before completing the Vary Configuration (VRYCFG) command.

Notes:

1. Specifying a wait time in the line description affects system IPL time, if ONLINE(*YES) is used, by the amount of time it takes to synchronously vary on the line or reach the wait-time value.
2. The time required to vary on a line is the time it takes to put tasks in place to manage the line, to activate the communications I/O processor (IOP) (including downloading the IOP model-unique Licensed Internal Code), and to establish communications with the data circuit-terminating equipment (DCE). Normal vary-on time ranges from 5 through 45 seconds, but can be longer, depending on the system, line protocol, and other factors.

LINESPEED

Specifies the line speed in bits per second (bps).

***SAME:** The value does not change.

***CALC:** The line speed is calculated by the system.

line-speed: Specify one of the following values (in bps) for the line speed:

600	4800	14400	56000
1200	7200	19200	57600
2400	9600	48000	64000

PKTMODE

Specifies whether to access the ISDN virtual circuit service.

***SAME:** The value does not change.

***NO:** The ISDN network is used to provide transparent access to an X.25 packet switched network external to the ISDN (Case A).

***YES:** The ISDN virtual circuit service is accessed (Case B).

INFTRFTYPE

Specifies the information transfer type. The information transfer type determines the layer 1 protocol.

***SAME:** The value does not change.

***UNRESTRICTED:** The data-channel traffic appears as digital information; no physical transformation is required and each B-channel operates at capacity (64k bps).

***V110:** The transfer type is V-series Recommendation 110. Each B-channel operates at 56k bps.

***IVTHDLC:** The transfer type is Inverted HDLC. Each B-channel operates at capacity, 64k bps.

EXTNETADR

Specifies whether network addressing is extended to permit the use of 17 characters in an address name.

***SAME:** The value does not change.

***NO:** Network addresses can be up to 15 characters.

***YES:** Network addresses can be up to 17 characters.

MAXFRAME

Specifies the maximum frame size that can be transmitted and received on this line description.

***SAME:** The value does not change.

max-frame: Specify one of the following values: 1024, 2048, 4096, 8192, or 16384.

DFTPKTSIZE

Specifies the default packet size used by the X.25 network.

Element 1: Transmit Packet Size

***SAME:** The value does not change.

transmit-packet-size: Specify a default packet size for transmission to all controllers that will attach to this line. The controllers can override this default with the DFTPKTSIZE parameter on the controller commands. The valid values for the packet size are 64, 128, 256, 512, 1024, 2048, and 4096.

Element 2: Receive Packet Size

***SAME:** The value does not change.

***TRANSMIT:** The value specified as the default packet size for transmission is used as the default for reception.

receive-packet-size: Specify a default packet size for reception from all controllers that will attach to this line. The controllers can override this default with the DFTPKTSIZE parameter on the controller commands. The valid values for the packet size are 64, 128, 256, 512, 1024, 2048, and 4096.

MAXPKTSIZE

Specifies the maximum packet size for transmission and reception on an X.25 network. The value specified must not be less than the default packet size specified.

Element 1: Maximum Transmit Packet Size

***SAME:** The value does not change.

***DFTPKTSIZE:** The maximum packet size for transmission is the same as that specified as the default packet size for transmission in the DFTPKTSIZE parameter in this command.

transmit-packet-size: Specify a packet size. The valid values for the packet size are 64, 128, 256, 512, 1024, 2048, and 4096.

Element 2: Maximum Receive Packet Size

***SAME:** The value does not change.

***DFTPKTSIZE:** The maximum packet size for reception is the same as that specified as the default package size for reception in the DFTPKTSIZE parameter in this command.

***TRANSMIT:** The value specified as the default packet size for transmission is used as the default for reception.

receive-packet-size: Specify a default packet size for reception from all controllers that will attach to this line. The controllers can override this default with the DFTPKTSIZE parameter on the controller commands.

MODULUS

Specifies the packet numbering used. The valid numbers are listed below.

***SAME:** The value does not change.

8: Modulus 8 packet numbering is used.

128: Modulus 128 packet numbering is used.

DFTWDWSIZE

Specifies the default window size used by the X.25 network.

Element 1: Transmit Window Size

***SAME:** The value does not change.

transmit-window-size: Specify the appropriate default window size. Valid values range from 1 through 7 for networks that use modulus 8 packet numbering. Valid values range from 1 through 15 for networks that use modulus 128 packet numbering.

Element 2: Receive Window Size

***SAME:** The value does not change.

***TRANSMIT:** The value specified as the default window size for transmission is used as the default for reception.

receive-window-size: Specify the appropriate default window size. Valid values range from 1 through 7 for networks that use modulus 8 packet numbering. Valid values range from 1 through 15 for networks that use modulus 128 packet numbering.

NETCTL

Specifies the name of an existing network controller.

SWTCTLLST

Specifies the names of up to 64 switched asynchronous X.25 controllers that can establish a connection with an X.25 switched virtual circuit (SVC). The controller descriptions must already exist, and must have been created by using the Create Controller Description (Async) (CRTCTLASC) command. Asynchronous X.25 controllers that are specified as dial-in can be connected to SVCIN or SVCBOTH logical channels. Attaching controllers that specify CNNNBR(*ANY) or ANSNBR(*ANY) may result in a reordering of this list.

***SAME:** The value does not change.

***NONE:** No switched controllers are specified.

switched-controller-name: Specify the switched controller name.

NETUSRID

Specifies, for switched lines, whether network user identification (NUI) information should be included in the facility field of call request packets sent on this line. The NUI facility identifies calling data terminal equipment (DTE) to the network.

***SAME:** The value does not change.

***NONE:** No NUI information is sent over this line.

network-user-identification: Specify the hexadecimal NUI that will sent. The system will insert the NUI facility code (X'C6') and the NUI length in the facility field of a call request packet.

IDLTMR

Specifies the maximum amount of time (in 0.1 second intervals) that the system waits for acknowledgment from the network for each frame sent before re-transmission.

Note: The IDLTMR value should be greater than or equal to the value of the following equation:

```
(2 * P + (MAXPKTSIZE*8)/LINESPEED + D) * 10
```

where "P" is the propagation delay (in seconds) of the medium that connects the user to the network, MAXPKTSIZE is the maximum transmit packet size, and "D" is the DCE (Data Circuit-terminating Equipment) processing overhead (in seconds). Contact the network provider for information about variables P and D.

***SAME:** The value does not change.

idle-timer: Specify a value ranging from 4 through 600. Each unit represents 0.1 seconds, which provides a timeout value ranging from 0.4 through 60 seconds.

FRAMERTY

Specifies the maximum number of retries attempted for various error conditions on the interface, such as link level time-outs, logical link level re-transmissions, and others.

***SAME:** The value does not change.

frame-retry: Specify a ranging value from 0 through 64.

ADRINSERT

Specifies whether the system inserts the local network address in CALL REQUEST or CALL ACCEPTED packets.

***SAME:** The value does not change.

***YES:** The local network address is inserted in packets.

***NO:** The local network address is not inserted in packets.

X25DCE

Specifies whether the system communicates by using the X.25 DCE support. It allows a system to communicate with another system without going through an X.25 network.

***SAME:** The value does not change.

***YES:** The AS/400 system does communicate by using the X.25 DCE support.

***NO:** The AS/400 system does not communicate by using the X.25 DCE support.

THRESHOLD

Specifies the temporary error threshold level being monitored by the system. A permanent error is reported only if the errors occurred consecutively and exceeded the retry limit.

Note: Specifying the THRESHOLD parameter affects all threshold errors. They cannot be specified individually.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** The error threshold is set at a minimum monitoring level.

***MED:** Error thresholding is set to a medium monitoring level.

***MAX:** The error threshold is set at a maximum monitoring level.

CNNNBR

Specifies, for switched lines, the telephone number used to establish a switched connection on this line description. The value specified on this parameter is sent to the autocall unit if automatic calling is used to establish a switched connection, or it is sent to the V.25 bis modem if V.25 bis automatic dialing is used to establish the connection. When manual dialing is used to establish a switched connection on the line, this number is displayed to the system operator. For X.25, this number is the remote data terminal equipment (DTE) address for the switched virtual circuit (SVC) provided by the X.25 network.

***SAME:** The value does not change.

connection-number: Specify the telephone number dialed to establish a switched connection on this line description.

CALLNBR

Specifies the local telephone number of the line used for the V.25 bis call request with identification (CRI) dial command. This parameter is used when the CRI function is needed for V.25 bis. When V.25 bis CRI dialing is used, the system takes the called (connection) number from the CNNNBR parameter of the controller description, adds a separator character (;), and concatenates the calling number at the end. Specify the calling number only if the modem and the network both support the CRI dial command.

***SAME:** The value does not change.

***NONE:** Specifies that the CRN (Call Request Normal) dial command is used by the V.25 bis line.

calling-number: Specify up to 32 characters that represent the local telephone number for V.25 bis CRI (Call Request with Identification) auto-dialing.

MODEM

Specifies the type of modem supported on the communications line. Refer to the modem information to select the appropriate value.

***SAME:** The value does not change.

***NORMAL:** No attempt is made to run diagnostic tests on the modem.

***V54:** Certain types of diagnostic tests (as defined by the CCITT recommendations) are run to the modem. The AS/400 system supports CCITT V.54 loop 3 (local loop back) and loop 2 (a remote loop back).

***IBMWRAP:** An IBM modem with wrap test capabilities is used on the communications line.

MODEMRATE

Specifies the speed at which the line operates if the modem has the data rate select feature.

Note: The user is responsible for ensuring that the line speed corresponds to the actual modem rate.

***SAME:** The value does not change.

***FULL:** The line operates at the full rate of the modem.

***HALF:** The line operates at half the full rate, or at the alternate rate, of the modem.

SWTCNN

Specifies whether the switched or switched network backup (SNBU) line is used for incoming calls, outgoing calls, or both incoming and outgoing calls.

***SAME:** The value does not change.

***BOTH:** The line is used for both incoming and outgoing calls.

***ANS:** The line is used for incoming calls only.

***DIAL:** The line is used for outgoing calls only.

CNNLSTOUT

Specifies, for ISDN switched connections, the name of a connection list object that contains the ISDN assigned numbers for a dial-out operation to the ISDN.

***SAME:** The value does not change.

connection-list-name: Specify the connection list object name.

CNNLSTOUTE

Specifies, for ISDN switched connections, the entry name from the connection list used to make a call to the ISDN. The connection list must be specified on the CNNLSTOUT parameter.

***SAME:** The value does not change.

connection-list-entry-name: Specify the entry name from the connection list.

CNNLSTIN

Specifies for ISDN switched connections the name of the connection list that is used to retrieve call information (or connection) for identifying authorized incoming calls.

***SAME:** The value does not change.

***NETATR:** The connection list used by this line description is taken from the list of system default network attributes that were identified at IPL (Initial Program Load). The Display Network Attributes (DSPNETA) command can be used to see the name of the connection list.

connection-list-name: Specify the name of the connection list used for this line description.

AUTOANS

Specifies, for switched or switched network backup (SNBU) lines, whether the system automatically answers a call from a remote system to establish the connection or whether the user must manually answer the call and place the modem in data mode.

***SAME:** The value does not change.

***YES:** The system automatically answers incoming calls.

***NO:** The system operator must manually answer incoming calls.

Note: *YES is valid only if the modem has the automatic answer feature.

AUTODIAL
Specifies, for switched or switched network backup (SNBU) lines, whether the system automatically calls a remote system to establish a connection or whether the system operator must manually place the call.

***SAME:** The value does not change.

***NO:** The AS/400 system does not automatically call a remote system.

***YES:** The AS/400 system automatically calls a remote system.

Note: *YES is only valid if the system is using an autocall unit or if the modem used is capable of calling though a command interface.

DIALCMD
Specifies the type of dial command used to establish a switched connection with a remote system.

***SAME:** The value does not change.

***NONE:** No dial command type is specified. An automatic call unit establishes the connection.

***V25BIS:** V.25 bis is a recommendation which allows the use of one physical interface for call establishment and data transmission. It is sometimes referred to as a serial automatic call interface because the digits are presented serially on the link from the system (DTE) to the modem (DCE).

CALLIMMED
Specifies, for switched (CNN(*SWTPP)) lines, whether a call (using the number specified by the CNNNBR parameter) should be made immediately after varying on the line description.

***SAME:** The value does not change.

***NO:** The call is not made immediately after varying on. It must be initiated by an application program.

***YES:** The call is made immediately after varying on.

AUTOCALL
Specifies, for switched lines, whether the line has an associated autocall unit which performs automatic calling to the remote system.

***SAME:** The value does not change.

***NO:** The switched line does not have an autocall unit.

***YES:** The switched line has an autocall unit.

ACRSRCNAME
Specifies the resource name that describes the automatic call unit port. The resource name consists of the input/output adapter (IOA) resource name and the port number on the IOA. For example, if the resource name of the IOA is LIN02 and the port is 1, the resource name is LIN021.

***SAME:** The value does not change.

autocall-resource-name: Specify the resource name that describes the automatic call unit port.

PREDIALDLY
Specifies how long to wait (in 0.5 second intervals) before dialing.

Note: This parameter is valid only for switched lines (CNN(*SWTPP)).

***SAME:** The length of time does not change.

predial-delay: Specify a value ranging from 1 through 254 in 0.5 second intervals, or specify 0 to indicate no delay.

REDIALDLY
Specifies the length of time (in 0.5-second intervals) to wait before re-dialing when the call attempt is unsuccessful. This parameter can be specified only if CNN(*SWTPP) is also specified.

***SAME:** The value does not change.

redial-delay: Specify a number ranging from 1 through 254 (measured in 0.5 second intervals), or specify 0 to indicate no delay.

DIALRTY
Specifies the number of re-dial attempts made by the system before considering the dialing unsuccessful.

Note: Dial retries can only be specified for switched lines (CNN(*SWTPP)).

***SAME:** The value does not change.

dial-retry: Specify a value ranging from 0 to 254 for the number of dial attempts.

SWTDSC
Specifies whether the switched connection on this line is disconnected when the last switched virtual circuit (SVC) is cleared.

***SAME:** The value does not change.

***YES:** The switched connection is disconnected when the last device is varied off.

***NO:** The switched connection is not disconnected when the last switched virtual circuit is cleared.

SWTDSCTMR
Specifies the settings for the timers that run when switched X.25 lines are disconnected from the network or remote system.

Element 1: Settings for the Minimum Connect Timer

***SAME:** The value does not change.

minimum-connect-timer: Specify the minimum length of time (in seconds) the system keeps the connection active. This timer is started when the connection is established. Valid values for the minimum length timer range from 0 to 65535 seconds.

Element 2: Settings for the Disconnect Delay Timer

disconnect-delay-timer: Specify the length of time (ranging from 0 through 65535 seconds) the system waits before disconnecting the switched line connection. The system disconnects the switched connection only when the following conditions occur:

- The line is idle
- The minimum connect timer has expired
- The disconnect delay timer has expired

DSRDRPTMR

Specifies the amount of time the system waits for the modem to exit the Data Set Ready (DSR) state before signaling an error.

***SAME:** The value does not change.

DSR-drop-timer: Specify a value ranging from 3 through 60 seconds.

AUTOANSTYP

Specifies the method the system uses to answer incoming calls.

***SAME:** The value does not change.

***DTR:** The system enters the Data Terminal Ready state, signals the modem to answer calls, and waits for the modem to enter the Data Set Ready (DSR) state.

***CDSTL:** The system enters the Connect Data Set to Line (CDSTL) state after monitoring the Ring Indicator to signal the modem to answer the call.

CTSTMR

Specifies the amount of time the system waits for the modem to enter or exit the Clear to Send (CTS) state before signaling an error.

***SAME:** The value does not change.

CTS-timer: Specify a value ranging from 10 through 60 seconds.

RMTANSTMR

Specifies the amount of time the system waits for the modem to enter the Data Set Ready (DSR) state after dialing before signaling an error.

***SAME:** The value does not change.

answer-timer: Specify a value ranging from 30 through 120 seconds.

LINKSPEED

Specifies the link speed in bits per second (bps). This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

***INTERFACE:** The following link speeds, based on the physical interface type, are used: 9600 bps for RS-232/V.24 and X.21bis/V.24, 48000 bps for V.35 and X.21bis/V.35, and 64000 bps for X.21 and RS-449/V.36.

***MIN:** A link speed of less than 1200 bps is used.

***MAX:** A link speed greater than 16M bps is used.

link-speed: Specify the link speed. Valid values are: 1200, 2400, 4800, 7200, 9600, 14400, 19200, 48000, 56000, 64000, 112000, 128000, 168000, 192000, 224000, 256000, 280000, 320000, 336000, 384000, 448000, 499000, 576000, 614000, 691000, 768000, 845000, 922000, 998000, 1075000, 1152000, 1229000, 1382000, 1536000, 1690000, 1843000, 1997000, 2048000, 4M, 10M, and 16M.

SWTNWISLCT

Specifies the method used to select network interfaces from the switched network interface list.

***SAME:** The value does not change.

***FIRST:** Selection begins with the first network interface specified in the switched network interface list.

***CALC:** The system calculates which network interface is selected.

COSTCNN

Specifies the relative cost of being connected on the line. This parameter is required only if APPN is used on the system.

***SAME:** The value does not change.

cost-per-connect-time: Specify a value ranging from 0 through 255.

COSTBYTE

Specifies the relative cost per byte for sending and receiving data on the line. This parameter is required only if APPN is used on the system.

***SAME:** The value does not change.

cost-per-byte: Specify a value ranging from 0 through 255.

SECURITY

Specifies the security level of the physical line. This parameter is used only if APPN is used on the system.

***SAME:** The value does not change.

***NONSECURE:** Normal priority is used.

***PKTSWTNET:** A packet switched network is used. Data does not always follow the same path through the network.

***UNDGRDCBL:** An underground cable is used.

***SECURECND:** A secure, unguarded conduit (for example, a pressurized pipe) is used.

***GUARDCND:** A guarded conduit, which is protected against physical tapping, is used.

***ENCRYPTED:** Data flowing on the line is encrypted.

***MAX:** A guarded conduit, protected against physical and radiation tapping, is used.

PRPDLY

Specifies the level of propagation delay on the line. This parameter is valid only if APPN is used on the system. The order of the values from shortest to longest delay is

*MIN, *LAN, *TELEPHONE, *PKTSWTNET, and *SATELLITE.

***SAME:** The value does not change.

***MIN:** The minimum propagation delay is used.

***LAN:** The local area network propagation delay is used.

***TELEPHONE:** The telephone propagation delay is used.

***PKTSWTNET:** The packet switched network propagation delay is used.

***SATELLITE:** The satellite propagation delay is used.

***MAX:** The maximum propagation delay is used.

USRDFN1

Specifies the first of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

user-defined-1: Specify a value ranging from 0 through 255.

USRDFN2

Specifies the second of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

user-defined-2: Specify a value ranging from 0 through 255.

USRDFN3

Specifies the third of the three user-defined fields. This field is used to describe unique characteristics of the line that is controlled. This parameter is valid only if APPN is used on the system.

***SAME:** The value does not change.

user-defined-3: Specify a value ranging from 0 through 255.

CMNRCYLMT

Specifies the number of recovery attempts made by the system before an inquiry message is sent to the system operator. Also specifies the time (in minutes) that must elapse before the system sends an inquiry message to the system operator indicating that the recovery attempt count limit is reached.

***SAME:** The value does not change.

***SYSVAL:** The recovery limits specified in the QCMNRCYLMT system value are used.

Element 1: Maximum Recovery Limit

count-limit: Specify the number of recovery attempts to be made. Valid values range from 0 through 99.

Element 2: Recovery Time Interval

time-interval: Specify the time interval (in minutes) at which the specified number of second-level recoveries are attempted. Valid values range from 0 through 120. If the value specified for *count-limit* is not 0, the value 0 specifies infinite recovery.

TEXT

Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

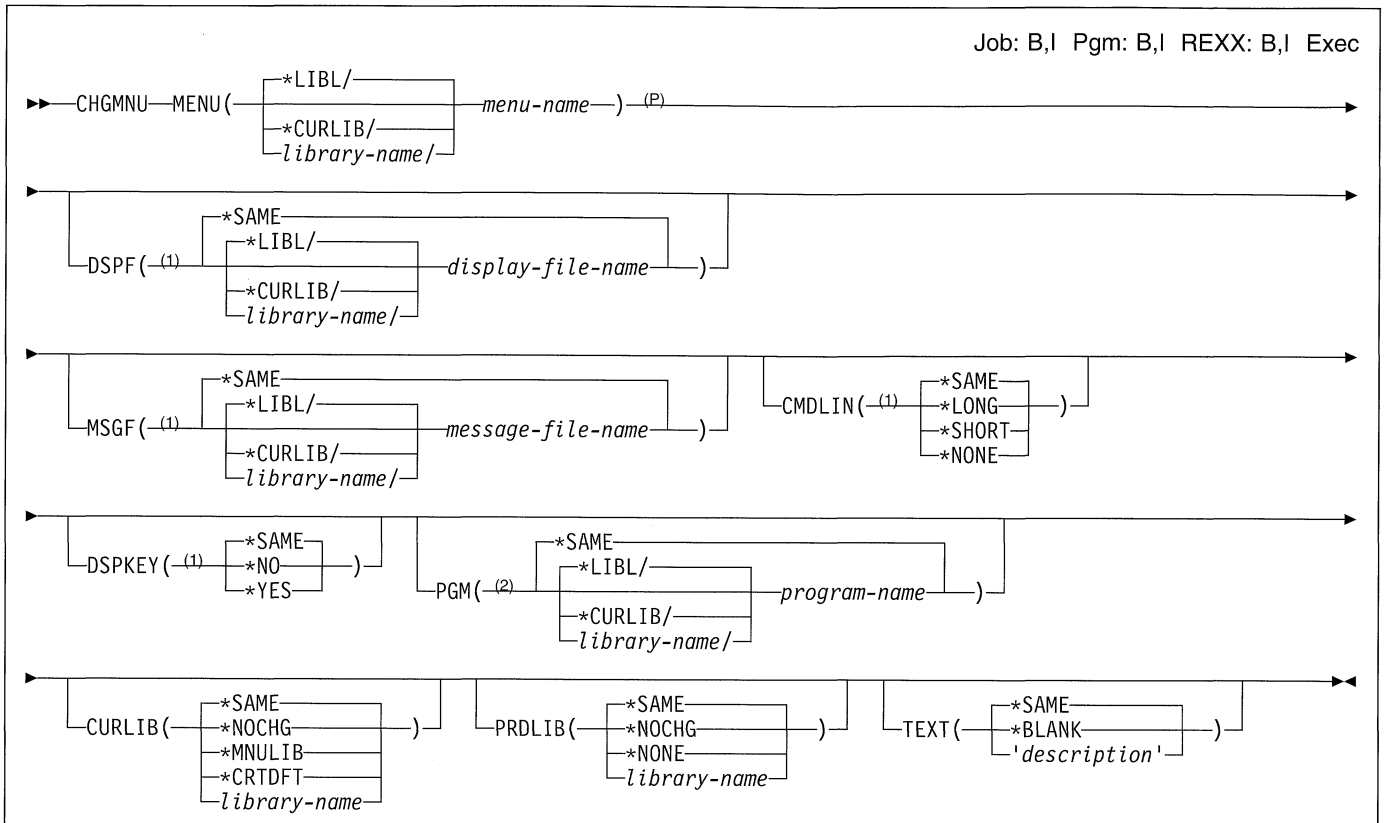
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGLINX25   LIND(X251)  LGLCHLE(*PROMPT)
  MAXPKTSIZE(512)
```

This command shows an interactive display listing all current logical channel entries and allows changes to be made to these entries. Maximum packet size will be changed to 512 bytes.

CHGMNU (Change Menu) Command



Notes:

1 The user can only specify the DSPF, MSGF, CMDLIN, and DSPKEY parameters if the menu was created with TYPE(*DSPF) specified on the CRTMNU command.

2 The user can only specify the PGM parameter if the menu was created with TYPE(*PGM) specified on the CRTMNU command.

P All parameters preceding this point can be specified in positional form.

Purpose

The Change Menu (CHGMNU) command can be used to change the following:

- Command Line
- Display function keys
- Current library and product library of a menu
- Display file name and library of a free format menu
- Message file name and library of a free format menu
- Program name and library of a program menu
- Text description of a menu

Restriction: The user must have *CHANGE authority for the menu object and *USE authority for the library.

Required Parameter

MENU

Specifies the qualified name of the menu being changed.

The name of the menu can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

menu-name: Specify the name of the menu being changed.

Optional Parameters

DSPF

Specifies the qualified name of the display file used for the menu. The display file must include one record format with the same name as the display file itself, called the menu format. Help formats may also be included in the file.

Help formats follow the naming convention **#Hxxyy**, where xx is the first menu option and yy is the last menu option to which the help format applies. (For example, #H0306 would apply to menu options 3 through 6.) **#H0000** designates the general help for the menu.

The display file must have a separate indicator area (INDARA keyword) and must not contain subfile descriptions.

This parameter can be specified only if the menu was created with TYPE(*DSPF) specified on the Create Menu (CRTMNU) command.

***SAME:** The display file name and library do not change.

The name of the display file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

display-file-name: Specify the name of the display file to use in the change.

MSGF

Specifies the qualified name of the message file that contains the commands to run when a menu option is selected. The MSGID's of the messages in this file are of the form USRxxxx, where xxxx is the number of the menu option that is typed on the command line.

This parameter can be specified only if the menu was created with TYPE(*DSPF) specified on the CRTMNU command.

***SAME:** The message file name and library do not change.

The name of the message file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

message-file-name: Specify the name of the message file to use.

CMDLIN

Specifies that the user can select a long command line, a short command line, or no command (an option line).

This parameter can be specified only if the menu was created with TYPE(*DSPF) specified on the CRTMNU command.

***SAME:** The value does not change.

***LONG:** A 153 byte long command line is used.

***SHORT:** A 73 byte long command line is used.

***NONE:** No command line is used. A 4-byte option line is used.

DSPKEY

Specifies whether the function key legend should appear at the bottom of the menu when it is shown.

This parameter can be specified only if the menu was created with TYPE(*DSPF) specified on the CRTMNU command.

***SAME:** The value does not change.

***NO:** The function key legend is not shown at the bottom of the display.

***YES:** The function key legend is shown at the bottom of the display.

PGM

Specifies the qualified name of the program to call when the menu is run.

This parameter can be specified only if the menu was created with TYPE(*PGM) specified on the CRTMNU command.

***SAME:** The program and library do not change.

The name of the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the program to call.

CURLIB

Specifies the name of the library being used as the current library during the processing of this command.

***SAME:** The value does not change.

***NOCHG:** The current library does not change for the processing of this command.

***MNULIB:** The current library is changed to be the library that contains the menu when the menu is shown. After exiting the menu, the current library is restored to the value it had before the menu was shown.

***CRTDFT:** There is no current library while the menu is shown. The current library that was active before the menu was shown (if any) is restored after exiting the menu.

library-name: Specify the name of the library used as the current library when the menu is shown. After exiting the menu, the current library is restored to the value it had before the menu was shown.

PRDLIB

Specifies the name of the library to use as the product library while the menu is shown.

***SAME:** The value does not change.

***NOCHG:** The product library is not changed when the menu is shown.

***NONE:** The product library entry in the library list is not used while the menu is shown.

library-name: Specify the name of the library to use as the product library while the menu is shown.

Note: After exiting the menu, the product library is restored to the value it had before the menu was shown.

TEXT

Specifies text that briefly describes the menu. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

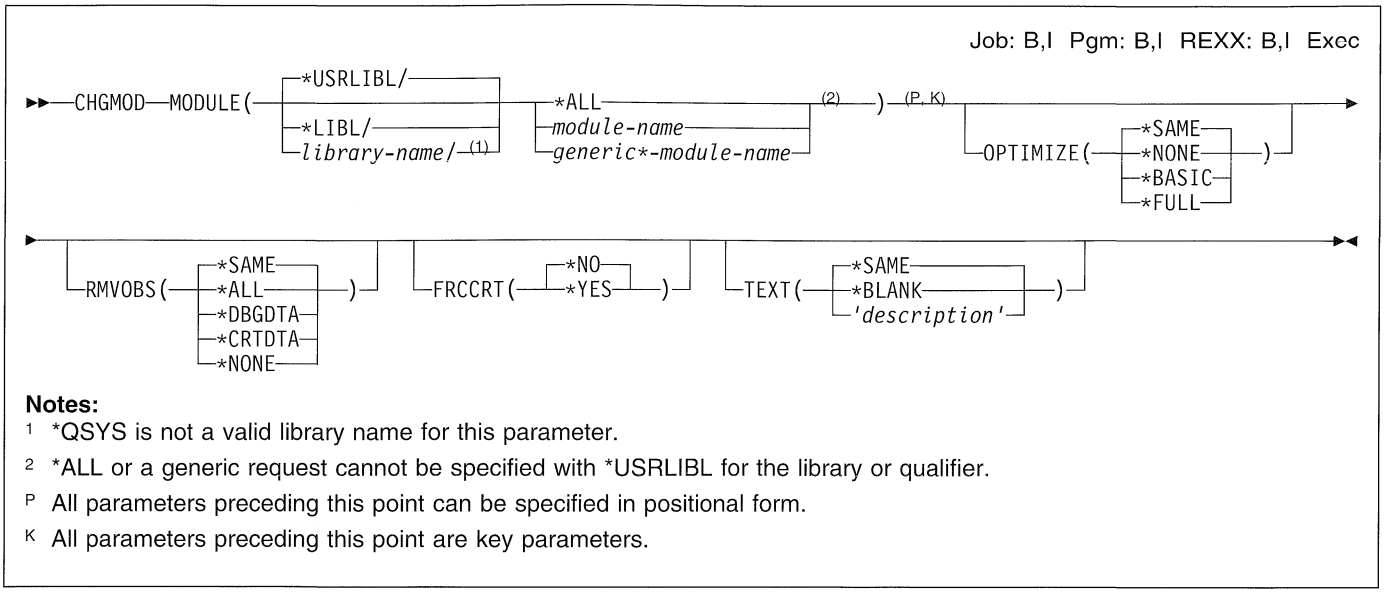
Example

```
CHGMNU   MENU(ARLIB/ARPERS)  PGM(ARLIB/PERSLIST)
```

This command changes the program that is called when a menu named ARPERS, located in library ARLIB, is run. The menu now calls a program called PERSLIST, also located in library ARLIB.

Note that the menu (ARPERS) must have been created with TYPE(*PGM) specified on the Create Menu (CRTMNU) command for the PGM parameter to be specified here.

CHGMOD (Change Module) Command



Purpose

The Change Module (CHGMOD) command changes the attributes of a module object without requiring the module to be recompiled.

Restrictions:

1. You must have *USE authority to the library where the module is located.
2. You must have *USE and object management authority for the module being changed.
3. You must have *ADD and *DELETE authority for the library in order to change the optimization attribute or to force module re-creation.
4. Modules in library QSYS cannot be changed.

Required Parameter

MODULE

Specifies the qualified name of the module for which attributes are changed.

The possible library values are:

***USRLIBL:** Only the libraries in the user portion of the job's library list are searched.

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

library-name: Specify the name of the library where the module is located.

***ALL:** All modules in the specified library to which the user has some authority are selected for change.

module-name: Specify the name of the module for which information is displayed.

generic-module-name:* Specify the generic name of the module being changed. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be changed only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

Optional Parameters

OPTIMIZE

Specifies the level at which the generated code in the module is used. Changing the optimization level of the module causes the system to re-create the module with the new optimization level.

***SAME:** The value does not change.

***NONE:** Additional optimization is not performed on the generated code. Variables can be displayed and modified while debugging.

***BASIC:** A specified amount of optimization is performed on the code. Variables can be displayed but not modified while debugging.

***FULL:** Maximum optimization is performed on the generated code. Variables can be displayed but not modified while debugging.

Note: The value of the variable displayed during debugging may not be the current value of the variable.

RMVOBS
Specifies whether the observable information associated with modules is removed.

***SAME:** The value does not change.

***ALL:** Remove the debug data (*DBGDTA) and information necessary to re-create the module (*CRTDTA).

***DBGDTA:** The information necessary to allow module debugging is removed. The module cannot be debugged when it is included in a program or service program object.

***CRTDTA:** The information necessary to allow module re-creation or to change the optimization level is removed.

***NONE:** Observable information associated with the module is not removed.

FRCCRT
Specifies whether module re-creation is forced.

***NO:** Module re-creation is not forced unless the optimization level of the module is changed. This allows the system to determine whether a change is required.

***YES:** Module re-creation is forced. If the data necessary to re-create the module object has been removed, then a change in optimization level or forced re-creation is not allowed.

TEXT
Specifies text that briefly describes the program and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Examples

Example 1: Optimizing a Module

```
CHGMOD   MODULE(XYZ/PERFMOD)  OPTIMIZE(*FULL)
```

This command optimizes module PERFMOD in library XYZ to ensure optimum performance.

Example 2: Allowing a Module to be Debugged

```
CHGMOD   MODULE(XYZ/NOOBSERV)  RMVOBS(*CRTDTA)
```

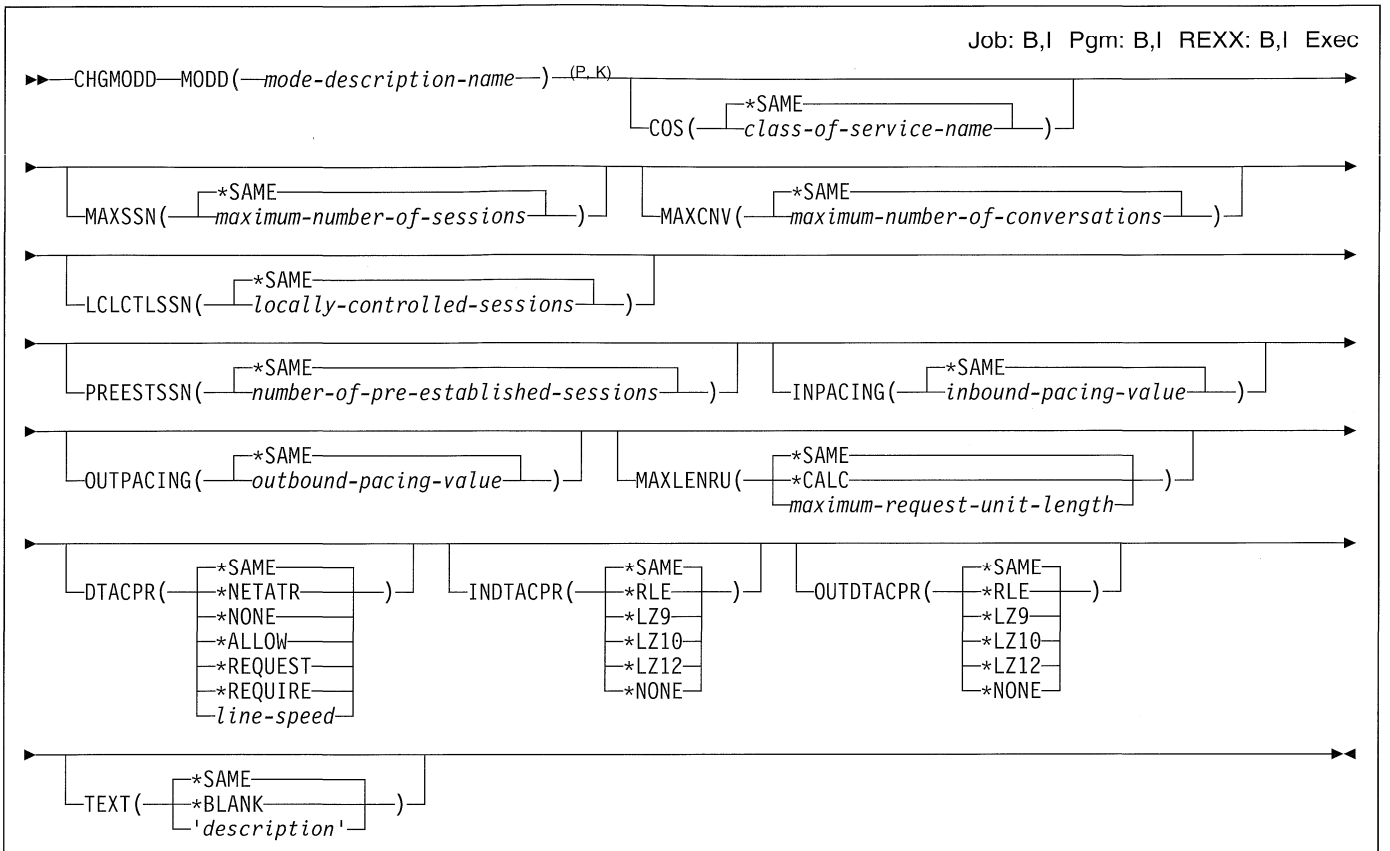
This command removes observability from module NOOBSERV in library XYZ and does not allow it to be re-created.

Example 3: Optimizing for Maximum Performance

```
CHGMOD   MODULE(XYZ/BEST)  RMVOBS(*ALL)
  OPTIMIZE(*FULL)
```

This command optimizes module BEST in library XYZ to ensure optimum performance and the smallest size.

CHGMODD (Change Mode Description) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Mode Description (CHGMODD) command changes a mode description.

Required Parameter

MODD

Specifies the name of the mode description being changed.

Optional Parameters

COS

Specifies the class-of-service associated with the mode.

***SAME:** The value does not change.

class-of-service-name: Specify the class-of-service name in the mode description. Valid class-of-service names are shown below:

- #CONNECT
- #BATCH
- #INTER
- #BATCHSC
- #INTERSC

MAXSSN

Specifies the maximum number of active sessions that are established for this mode. This number must be greater than or equal to the number of locally controlled sessions on the local system (as specified in the LCLCTLSSN parameter) plus the number of locally controlled sessions specified at the remote location.

***SAME:** The value does not change.

maximum-number-of-sessions: Specify a value ranging from 1 through 512 for the maximum number of active sessions.

MAXCNV

Specifies the maximum number of active conversations allowed by this mode. The maximum number of active conversations is the sum of synchronous and asynchronous conversations; this value must be greater than or equal to the value specified by the MAXSSN parameter. A synchronous conversation is a conversation in which both the source and target programs are communicating. An asynchronous conversation is a conversation in

which the source program has detached from the conversation, but there is still data to be read by the target program.

***SAME:** The value does not change.

maximum-number-of-conversations: Specify a value ranging from 1 through 512 for the maximum number of conversations.

LCLCTLSSN

Specifies the minimum number of active locally controlled sessions that can be established for this mode. This value must be less than or equal to the value specified in the MAXSSN parameter.

***SAME:** The value does not change.

locally-controlled-sessions: Specify a value ranging from 0 through 512 for the minimum number of locally controlled sessions.

PREESTSSN

Specifies the maximum number of locally controlled pre-established sessions (established when the mode is started). Additional sessions are established when required up to the value specified in the MAXSSN parameter; this value must be less than or equal to the value specified in the LCLCTLSSN parameter.

***SAME:** The value does not change.

number-of-pre-established-sessions: Specify a value ranging from 0 through 512 for the maximum number of locally controlled sessions established when the mode is started.

INPACING

Specifies the Systems Network Architecture (SNA) pacing value used to schedule the incoming request/response units (RUs).

***SAME:** The value does not change.

inbound-pacing-value: Specify a value ranging from 0 through 63 for the limiting value.

OUTPACING

Specifies the Systems Network Architecture (SNA) pacing value used for outgoing request/response units (RUs).

***SAME:** The value does not change.

outbound-pacing-value: Specify a value ranging from 0 through 63 as the limiting value.

MAXLENRU

Specifies the maximum request unit (RU) length (in bytes) allowed.

***SAME:** The value does not change.

***CALC:** The system determines the value to use.

maximum-request-unit-length: Specify a value ranging from 241 through 16384 as the maximum length for incoming request units.

Note: *CALC is the recommended value. Some other common values are:

SDLC lines:	256, 512, 1024, 2048
Token-ring network lines:	256, 512, 1024, 1985
X.25 (QLLC) lines:	247, 503, 1015
X.25 (ELLC) lines:	241, 497, 1009

DTACPR

Specifies whether data compression is used.

***SAME:** The value does not change.

***NETATR:** The value from the DTACPR network attributes is used.

***NONE:** Compression is not allowed on the session.

***ALLOW:** Data compression is allowed on the session by the local system if requested by a remote system. The local system does not request compression.

If data compression is requested by the remote system, the data compression levels used by the session are the lower of the requested levels and the levels specified on the INDTACPR and OUTDTACPR parameters.

***REQUEST:** Data compression is requested on the session by the local system. However, the request can be refused or changed to lower compression levels by the remote system. Data compression is allowed on the session if requested by the remote system. The requested compression levels for inbound and outbound data are the levels specified for the INDTACPR and OUTDTACPR parameters.

If data compression is requested by the remote system, the data compression levels used by the session are the lower of the requested levels and the levels specified on the INDTACPR and OUTDTACPR parameters.

***REQUIRE:** Data compression is required on the session. If the remote system does not accept the local system's exact required levels of compression, the session is not established.

The data compression levels that the local system require are the levels specified on the INDTACPR and OUTDTACPR parameters.

line-speed: Specify the maximum line speed at which data is compressed. If the line speed of the link used by the session is less than or equal to this specified line speed, data compression is used for the session as if *REQUEST is specified. Otherwise, compression is used for the session as if *ALLOW is specified. Valid values range from 1 through 2147483647 in bits per second (bps).

INDTACPR

Specifies the desired level of compression for inbound data. No data compression occurs if DTACPR(*NONE) is specified.

Note: Adaptive dictionary-based compression is a dynamic compression algorithm, similar to

Lempel-Ziv, that compresses previously seen strings to 9-, 10-, and 12-bit codes. This algorithm is referred to as LZ in the following parameters.

***SAME:** The value does not change.

***RLE:** The Run Length Encoding (RLE) algorithm is used. RLE substitutes a 1- or 2-byte sequence in the data stream for each repeated occurrence of the same character. This algorithm requires no storage and less processing time than the other options.

***LZ9:** The LZ algorithm with the 9-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require storage and extra processing time. The LZ9 requires the least storage and processing time of the LZ algorithms; however, it compresses the data stream the least.

***LZ10:** The LZ algorithm with the 10-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require storage and extra processing time. The LZ10 table algorithm requires more storage and processing time than the LZ9, but less than the LZ12. The LZ10 compresses the data stream more than the LZ9, but less than the LZ12.

***LZ12:** The LZ algorithm with the 12-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require storage and extra processing time. The LZ12 requires the most storage and processing time of the LZ algorithms; however, it compresses the data stream the most.

***NONE:** No compression occurs.

OUTDTACPR

Specifies the desired level of compression for outbound data. No data compression occurs if DTACPR(*NONE) is specified.

***SAME:** The value does not change.

***RLE:** The Run Length Encoding (RLE) algorithm is used. RLE substitutes a 1- or 2-byte sequence in the data stream for each repeated occurrence of the same character. This algorithm requires no storage and less processing time than the other options.

***LZ9:** The LZ algorithm with the 9-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require storage and extra processing time. The LZ9 requires the least storage and processing time of the LZ algorithms; however, it compresses the data stream the least.

***LZ10:** The LZ algorithm with the 10-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require storage and extra processing time. The LZ10 table algorithm requires more storage and processing time than the LZ9, but less than the LZ12. The LZ10 compresses the data stream more than the LZ9, but less than the LZ12.

***LZ12:** The LZ algorithm with the 12-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require storage and extra processing time. The LZ12 requires the most storage and processing time of the LZ algorithms; however, it compresses the data stream the most.

***NONE:** No compression occurs.

TEXT

Specifies text that briefly describes the mode description. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

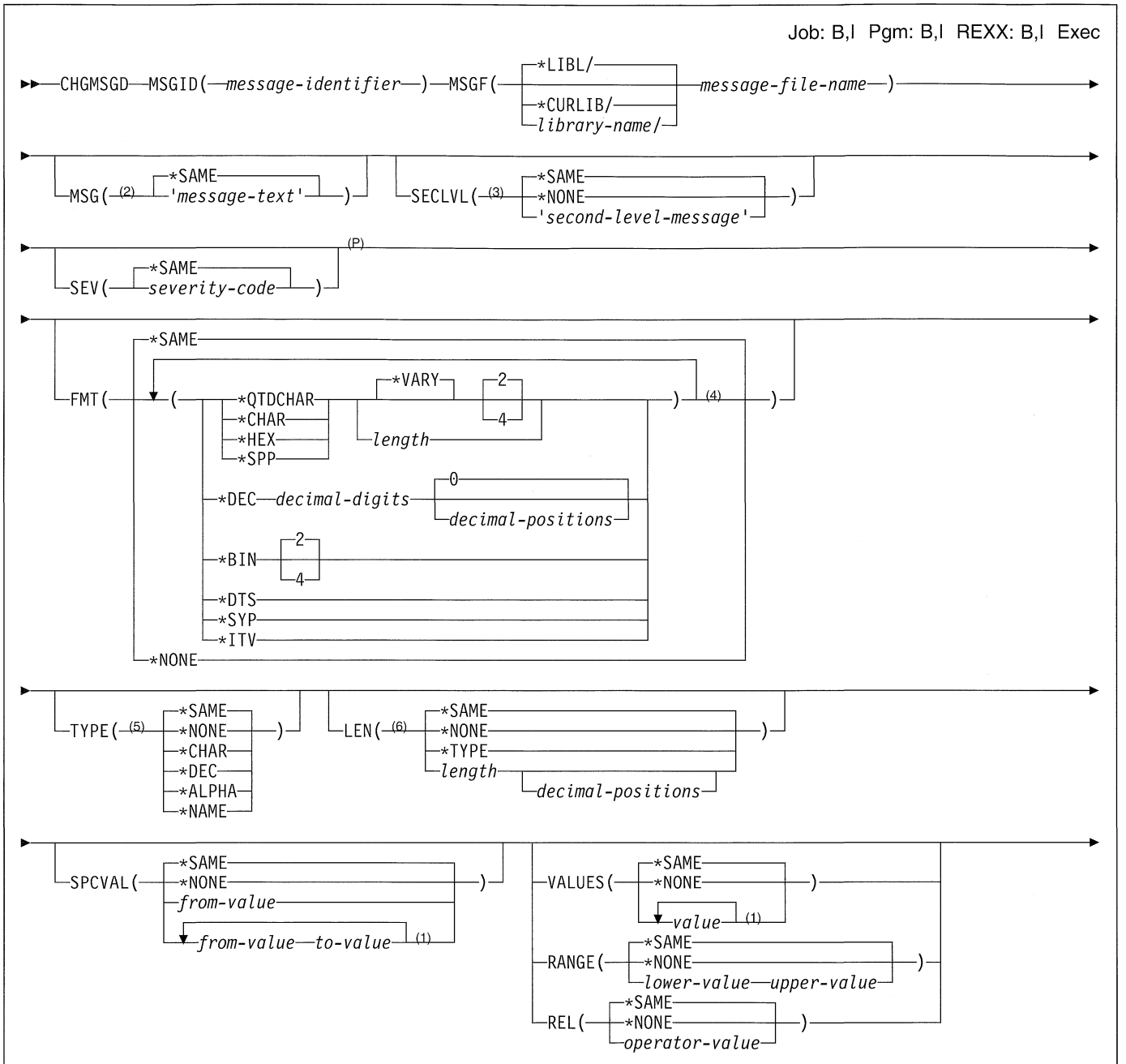
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGMODD   MODD(MODE1)  PREESTSSN(3)
```

This command changes the number of pre-established sessions for MODE1 to three.

CHGMSGD (Change Message Description) Command



Notes:

1 A maximum of 20 repetitions

2 No more than 132 characters can be specified.

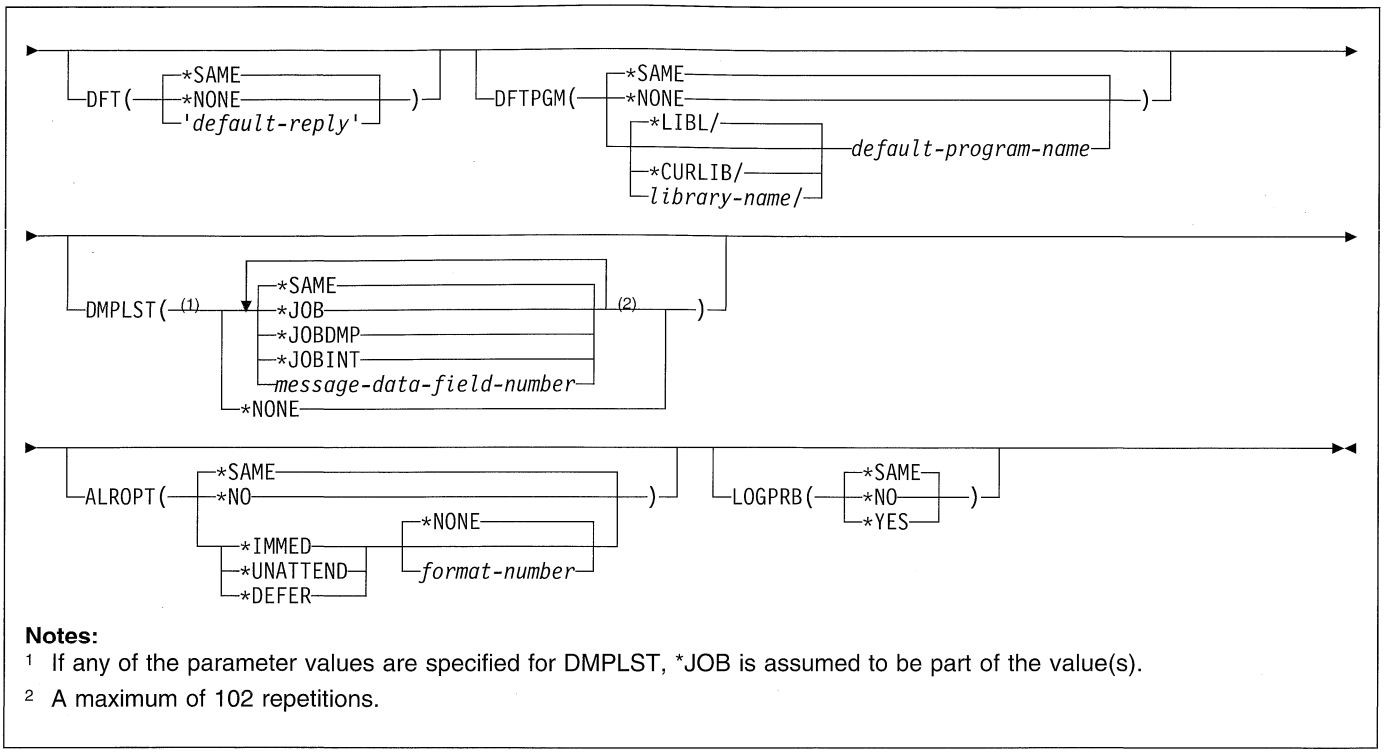
3 No more than 3000 characters can be specified.

P All parameters preceding this point can be specified in positional form.

4 A maximum of 99 repetitions.

5 If LEN is specified as *NONE, TYPE must also be specified as *NONE.

6 If TYPE is specified as *NONE, LEN must also be specified as *NONE.



Notes:
1 If any of the parameter values are specified for DMPLST, *JOB is assumed to be part of the value(s).
2 A maximum of 102 repetitions.

Purpose

The Change Message Description (CHGMSGD) command changes an existing message description stored in a message file and stores those changes in that message file for later use. The message description remains in the message file until the file is deleted, until the Remove Message Description (RMVMSGD) command is used to remove the message from the file, or until the message is changed again by using the CHGMSGD command.

Note: If the message and its second-level text exceeds 512 characters, it will not fit in the prompt field. In this case, enter the command on the Command Entry panel or in a CL program.

Caution: Specifying the alert option in the range CPx7B00 to CPx7BFF (messages sent from the alert process) may cause unpredictable results.

Required Parameters

MSGID

Specifies the message identifier of the message being changed. The message identifier must be seven characters long and in the format, pppnnnn.

The first three characters (ppp) must be a code consisting of one alphabetic character followed by two alphanumeric (alphabetic or decimal) characters; the last four characters (nnnn) may consist of the decimal numbers 0 through 9 and the characters A through F.

MSGF

Specifies the qualified name of the message file where the message being changed is stored. This command ignores any message file overrides in effect for the job.

The name of the message file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

message-file-name: Specify the name of the message file to use.

Optional Parameters

MSG

Specifies the first-level message text of the message being changed.

***SAME:** The value does not change.

'message-text': Specify the message text that is initially displayed, printed, or sent to a program or log. Up to maximum of 132 characters (enclosed in apostrophes) can be specified, but the work station display size may cause additional limitations.

Note: If the message text is changed, the entire original message text is replaced with the specified change.

One or more substitution variables can be embedded in the message text string to indicate positional replacement fields. These replacement fields allow variable data to be substituted in the message by the program before the message is sent. The rules below must be followed when variables are used.

- Variables must be specified in the form &n, where n is a 1- or 2-digit (1-99) number identifying the data field that is substituted.
- Variables can be preceded by any alphanumeric character (including blanks). For example, the variables shown in the message below are valid.

  ```
  Command&34&72 &2 &99help
  ```

- Variables can be followed by any non-numeric character (the character following the variable cannot be digits 0-9). For example, the variables shown below are not valid.

  ```
  Command&345 &244 &999help
  ```

- The variables can be enclosed in apostrophes if only the variables themselves make up the message. For example, to show a two-part decimal value, the message '&1.&2' can be specified.)
- Variables do not have to be in ascending or descending sequence.

Note: The data fields are described positionally in the FMT parameter and are specified positionally in the MSGDTA parameter of the Send Program Message (SNDPGMMSG) command and the Send User Message (SNDUSRMSG) command. More details on substituting data fields in message text are in the *CL Programmer's Guide*.

Double-Byte Character Set Considerations

When entering double-byte characters on this parameter, several combinations of characters may cause errors to occur on this command. If the double-byte characters contain the string, X'50Fn' (where n is a 1-digit number, 0-9), error messages CPF2424 or CPF2431 may result. Examples are: X'50F0', X'50F4', X'50F9'.

SECLVL

Specifies second-level message text being changed. Second-level message text can also be written to the job log, if *SECLVL is specified on the LOG parameter of the job commands.

***SAME:** The value does not change.

***NONE:** There is to be no second-level message text for this message description. Second-level message text in the original message description is removed.

'second-level-message': Specify the second-level message text. Up to 3000 characters (enclosed in apostrophes) can be specified, but display limitations must be considered. One or more substitution variables can be embedded in the second-level message text, as described in the MSG parameter. If second-level message text is changed, the entire second-level message text is replaced with the specified change.

Second-level message text can be formatted for the work station using three format control characters:

- **&N:** Forces the text to a new line (column 2). If the text is longer than one line, the next lines are indented to column 4 until the end of the text or until another format control character is found.
- **&P:** Forces the text to a new line, indented to column 6. If the text is longer than one line, the next lines start in column 4 until the end of the text or until another format control character is found.
- **&B:** Forces the text to a new line, starting in column 4. If the text is longer than one line, the next lines are indented to column 6 until the end of the text or until another format control character is found.

Double-Byte Character Set Considerations

When entering double-byte characters on this parameter, several combinations of characters may cause errors to occur on this command. If the double-byte characters contain the string, X'50Fn' (where n is a 1-digit number, 0-9), error messages CPF2424 or CPF2431 may result. Examples are: X'50F0', X'50F4', X'50F9'.

SEV

Specifies the severity code of the message. The severity code indicates the severity level of the condition that causes the message to be sent. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

severity-code: Specify a value, 00 through 99, for the severity level of this message. The assigned code for the message should correspond to the predefined IBM severity codes. Any 2-digit value can be specified, even if no severity code (predefined or user-defined) has been defined for it.

FMT

Specifies the formats of from 1 through 99 message data fields being changed. Each field is described in this parameter by a list of attributes. All 99 of the message data fields can be used as substitution values in the first-level and second-level messages defined in this message description. They can also be specified in the DMPLST parameter of this command. When specified in the MSGDTA parameter of the Send Program Message (SNDPGMMSG) or Send User Message (SNDUSRMSG) commands, the data fields must be concatenated to form one character string of no more than 512 characters and must match the format and sequence specified here.

Note: If any of the previously defined formats are being changed, all existing formats must be included in the FMT parameter. For example, if seven

formats had been previously defined and now the third of the seven formats is to be changed from *CHAR 24 to *HEX 8, all seven of the formats (including their types and lengths) must be included in the FMT parameter.

***SAME:** The value does not change.

***NONE:** Either no format is being described for the message fields, or the original formats are removed. If *NONE is specified, no references can be made to message data fields in the MSG, SECLVL, or DMPLST parameters.

Note: If FMT was originally specified, but now FMT(*NONE) is specified, all references to those formats must be removed from the first-level and second-level messages and from the dump list.

type (length[decimal-positions]): Specify a list of attributes that define each message data field (up to 99 fields) in this message description. These attributes specify the type of data in the field, the total length of the field, and, optionally, the number of decimal digits to the right of the decimal point. Certain data types do not require a length field. Boundary alignment requirements must be considered (for example, pointers are always aligned on 16-byte boundaries). Fields &1 through &99 can be defined in and can appear in, the message text. Fields beyond that number can appear only in the dump list.

Type of Message Data:

The first value specifies the type of data the substitution field contains and how the data is formatted in the message text. The contents of the second and third values vary depending on the message type specified. One of the following types can be specified for each field described by this parameter:

***QTDCHAR:** A character string formatted (by the Operating System/400) with enclosing apostrophes (for example, 'Monday, the 1st') is specified.

***CHAR:** A character string formatted without enclosing apostrophes is used. An alphanumeric string is used, for example, to specify a name (BOB). Trailing blanks are truncated.

***HEX:** A string of bytes formatted as a hexadecimal value (for example, X'C0F4') is specified.

***SPP:** A 16-byte space pointer to data in a space object is specified. When referred to in the DMPLST parameter, the data in the space object (from the offset indicated by the pointer) for the length specified is dumped. *SPP is not valid as a replacement field in message text.

***DEC:** A packed decimal number (for example, X'058C') that is formatted in the message as a signed decimal value with a decimal point (for example, 58.) is specified. Values for length (required) and decimal positions (optional) specified *DEC indicate the number of decimal digits and the number of digits to the right of the decimal point. If the number of decimal positions is not specified, zero is assumed.

***BIN:** A binary value that is either 2 or 4 bytes long (for example, B'0000 0000 0011 1010'), which is formatted in the message as a signed decimal value (for example, 58), can be specified.

The following formats are valid only in IBM-provided message descriptions and should not be used for other messages.

***DTS:** An 8-byte field that contains a system-date time-stamp is specified. The date in the output message is in the format specified by the system values QDATFMT and QDATSEP. The time is formatted as hh:mm:ss.

***SYP:** A 16-byte system pointer to a system object is specified. When referred to in message text, the simple name of the system object is formatted as described in the name type, *CHAR. When referred to by the DMPLST parameter, the object itself is dumped.

***ITV:** An 8-byte field that contains a time interval is specified. The time is formatted in the output message in the form of seconds.

Length of Message Data:

After the type specification, a second value (length) can be specified to indicate the number of characters or digits that are passed in the message data. How the second value is used depends on the type specified in the first value.

- If a length is not specified for *QTDCHAR, *CHAR, *HEX, or *SPP, then *VARY is assumed for the length. If *VARY is assumed, the message data field passed by the SNDPGMMSG or SNDUSRMSG commands must be preceded by a 2-byte or 4-byte binary field that indicates the actual number of bytes of data being passed. However, when *SPP is specified, the first bytes pointed to by the space pointer contain the field length. Therefore, the 2- or 4-byte field must precede the data pointed to by the space pointer, and must not precede the space pointer that is passed as part of the message data.
- If the type *DEC is specified, the total number of decimal digits (including the fraction) must be specified as the second value; the number of digits in the fraction can be specified optionally as the third value.
- If the type *BIN is specified, the message data field can be only 2 or 4 bytes long. The default is 2 bytes.

Length Field Size/Decimal Positions:

The third value is used in one of two ways, depending on the type specified in the first value: (1) if *QTDCHAR, *CHAR, *HEX, or *SPP is specified, and if *VARY is specified or assumed for the second value, the third value is used with *VARY to indicate the size of the length field actually passed. The third value can be

either 2 or 4, which is the number of bytes specifying the length (in binary) of the passed value; (2) if *DEC is specified, the third value indicates the number of decimal positions in the decimal value. If not specified for a decimal substitution value, the default is 0 decimal positions.

Note: If an object has been damaged or deleted, the substitution variable, when displayed, is not replaced by the name of the object. Instead, the object appears as &n (where n = number).

Reply Validity Specification Parameters

If the message is sent as an inquiry message or as a notify message (specified by MSGTYPE(*INQ) or MSGTYPE(NOTIFY) on the SNDPGMMSG command) and a reply is expected, seven parameters can be used to specify some requirements that validate the reply received. The seven validity-checking parameters are: TYPE, LEN, VALUES, SPCVAL, RANGE, REL, and DFT.

These parameters are not necessary for a message to allow a reply, but they can be used to define valid replies made to the message. The VALUES, RANGE, and REL are mutually exclusive; only one of them can be specified in this command.

Note: If the reply type or length is changed, and if VALUES, RANGE, or REL had been previously specified, the existing VALUES, RANGE, REL, SPCVAL and DFT must also be changed to be compatible with the new reply type and/or length. If the reply type is changed, LEN must be changed also. If the reply type is changed to *NONE, then LEN and (if they were coded previously) VALUES, SPCVAL, RANGE, REL, and DFT must be coded as *NONE.

TYPE

Specifies, only if the message is sent as an inquiry or notify message, the type of valid reply to this message.

***SAME:** The value does not change.

***NONE:** Reply message validity is not checked. Existing reply message type specifications are removed. LEN(*NONE) also must be specified.

***CHAR:** Any character string is a valid reply message type specification. If it is a quoted character string, the apostrophes are passed as part of the character string.

***DEC:** Only a decimal number is a valid reply message type.

***ALPHA:** Only an alphabetic (A through Z, $, #, and @) character string is a valid reply message type specification. Blanks are not allowed.

***NAME:** Only a simple name is a valid reply message type specification. The name does not have to be a OS/400 system object name, but must start with an alphabetic character; the remaining characters must be alphanumeric. If all reply characters are alphabetic (A-Z), the reply is converted to uppercase characters.

LEN

Specifies, only if the message is sent as an inquiry or notify message, the maximum reply length.

***SAME:** The value does not change.

***NONE:** The validity of the reply message is not checked. The existing LEN specification, if any, is removed and TYPE(*NONE) must also be specified.

***TYPE:** The maximum length is determined by the type of reply specified on the TYPE parameter. The maximum length for each type of reply is as follows:

- Up to 132 characters can be specified for types *CHAR and *ALPHA. If additional validity checking is being performed (for example, if VALUES, RANGE, REL, or SPCVAL are specified), the maximum length allowed for *CHAR and *ALPHA is 32 characters.
- Up to 15 digits can be specified for *DEC, of which up to 9 digits can be to the right of the decimal point.
- Up to 10 alphanumeric characters can be specified for *NAME.

length (decimal-positions): Specify the maximum reply length. The length specified here cannot exceed the maximums shown above. If the reply type is a decimal value, the number of decimal positions can be optionally specified; if a decimal is not specified, zero decimal positions are assumed.

VALUES

Specifies, only if the message is sent as an inquiry or notify message, a list of values, one of which is received as a valid reply. No more than 20 values can be specified in the list. Each value in the list must meet the requirements specified for message replies by the TYPE and LEN parameters. If the VALUES parameter is specified, the RANGE and REL parameters cannot be specified.

***SAME:** The value does not change.

***NONE:** No list of reply values is specified. The reply can have any value that is consistent with the other validity-checking parameters. Any existing VALUES are removed.

value: Specify up to 20 values. To be valid, these must match a reply value sent in response to the message defined in this message description. The maximum length of each value is 32 characters.

SPCVAL

Specifies, only if the message is sent as an inquiry or notify message, a list of up to 20 sets of special values of which one set (the from-value matched by the sent reply) is used as the reply. These values are special in that they may not meet all of the validity checking specifications given in the other reply-oriented parameters. The reply message sent is compared to the from-value in each set, if a match is found, and a to-value was

specified in that set, the to-value is sent as the reply. If no to-value was specified, the from-value is sent as the reply. If the reply sent does not match any from-value, then the reply is validity-checked by the specifications in the other reply-oriented parameters.

***SAME:** The value does not change.

***NONE:** No special values are specified for the replies to this message. Any existing special values are removed from the message description.

from-value (to-value): Specify up to 20 sets of values to determine the reply sent to the sender of the message. Each set must have a from-value with which the reply is compared, and an optional to-value to be sent as the reply (if its from-value matches the reply).

RANGE

Specifies, only if the message is sent as an inquiry or notify message, the lower and upper value limits for valid replies to this message. These values must meet the requirements specified for replies by the TYPE and LEN parameters, and both values must be of the same type. If both values are not of the same length, the shorter value is padded on the right with blanks. For type *CHAR and *ALPHA replies, the reply is padded on the right with blanks or truncated on the right (to the length of the specified values) before the value range is validity-checked. If RANGE is specified, the VALUES and REL parameters cannot be specified.

***SAME:** The value does not change.

***NONE:** No range values are specified for the replies to this message. Any existing range values are removed from the message description.

lower-value upper-value: Specify the lower and upper limit values for valid replies to this message.

REL

Specifies, only if the message is sent as an inquiry or notify message, the relationship that must exist for a reply to be valid. The value specified must meet the requirements specified for replies by the TYPE and LEN parameters. For replies of the types *CHAR and *ALPHA, the reply is padded on the right with blanks or truncated on the right to match the length of the value specified, before the system performs the test on the reply value that is sent.

***SAME:** The value does not change.

***NONE:** No relationship is specified for replies to this message. Existing relationship specifications are removed from the message description.

operator-value: Specify one of the relational operators and the value against which the reply message is checked. If the reply is valid in the relational test, it is sent to the sender of the message. The relational operators that can be entered are:

*LT	Less than
*LE	Less than or equal to
*GT	Greater than
*GE	Greater than or equal to
*EQ	Equal to
*NL	Not less than
*NG	Not greater than
*NE	Not equal to

Note: If VALUES, RANGE, or REL had been specified on the existing message, and they are being changed to another type of reply validity-checking, then the existing check must be removed by specifying *NONE. For example, if VALUES had been specified originally, but now the user wants to specify a RANGE, then VALUES(*NONE) and RANGE(*lower-value upper-value*) must be specified in the CHGMSGD command.

DFT

Specifies, only if the message is sent as an inquiry or notify message, the default reply (enclosed in apostrophes, if it contains special characters) used when the receiver of the message has indicated that all messages to him or her are to use default replies, or when a message is deleted from a message queue and no reply was specified. The default reply can also be used to answer unmonitored notify messages. The default reply must meet the requirements specified for replies by the validity-checking parameters.

***SAME:** The value does not change.

***NONE:** No default reply is specified. Existing default replies are removed.

'default-reply': Specify the reply, enclosed in apostrophes if it contains special characters, to use as the default reply.

DFTPGM

Specifies the qualified name of the default program (if any) that takes default action when this message is sent as an escape message to a program that is not monitoring for it. This parameter is ignored if the message is not sent as an escape message. If it is sent as an escape message, the following parameters are passed to the default program:

- Program message queue name (10 characters). The name of the program message queue to which the message was sent. This is the same name as that of the program that did not monitor for the message.
- Message reference key (4 characters). The message reference key of the escape message on the program message queue.

***SAME:** The value does not change.

***NONE:** No default program is specified for this message. Any existing default program is removed from the message description.

The name of the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

default-program-name: Specify the name of the default program that is called when an escape message is sent.

DMPLST

Specifies the data that is dumped when this message is sent as an escape or notify message to a program that is not monitoring for it. This parameter can specify that data related to the job be dumped, that data from message data fields be dumped, or that a combination of these be dumped. When data from message data fields is dumped, this parameter specifies one or more numbers that positionally identify the data fields being dumped.

The system objects indicated by system pointers are dumped. The data in a space object, indicated by a space pointer, is dumped starting from the offset indicated by the space pointer for the length indicated in the field description. The standard job dump can also be requested.

Notes:

1. If any of these values are specified for DMPLST, *JOB is assumed to be part of the values. For example, DMPLST (1 2 *JOBDMP) gives the same result as DMPLST(*JOB 1 2 *JOBDMP).
2. Any value specified for the DMPLST parameter may be overridden by the QSRVDMP system value. More information is in the chapter on system values in the *Work Management Guide*.
3. The program receiving the unmonitored message must have a name starting with 'Q' or the message severity must be 50 or higher.
4. The user of the job in which the dump is specified must be authorized to the dump command requested on this parameter.

***SAME:** The value does not change.

***JOB:** This value has the same effect as specifying JOB(*) and OUTPUT(*PRINT) on the Display Job (DSPJOB) command. See the DSPJOB command description for more information.

***JOBDMP:** The data areas of the job are dumped as specified by the Dump Job (DMPJOB) command. *JOBDMP can be specified either by itself, with *JOB, with *JOBINT, or with a list of message data field numbers.

***JOBINT:** The internal machine data structures related to the job processing are dumped to the machine error log as specified by the DMPJOBINT command. *JOBINT can be specified either by itself, with *JOBDMP, with *JOB, or with a list of message data field numbers.

message-data-field-number: Specify the numbers of the message data fields that identify the data dumped when this escape message is sent but not monitored. As many as 99 data field numbers can be specified in the list. In addition, the list can contain the values *JOB and *JOBINT.

Single Value

***NONE:** There is no dump list for this message. Any existing dump list is removed from the message description.

ALROPT

Specifies the alert option associated with messages sent to the QHST log and the system operator message queue. More information is in the *Alerts and DSNX Guide*.

***SAME:** The value does not change.

***NO:** No alert is sent.

Element 1: When an Alert is Sent

***IMMED:** An alert is sent immediately, simultaneously sending the message to QHST.

***UNATTEND:** An alert is sent immediately only when the system is running in the unattended mode.

The system is considered to be unattended when the Alert Status (ALRSTS) network attribute is set to *UNATTEND.

***DEFER:** The alert is sent after local problem analysis.

*DEFER should be specified only for messages against which problem analysis can be run. An alert is sent at the first exit from problem analysis for the problem referred to by the message.

All alerts set to *DEFER are treated as *IMMED if:

- The system is running in the unattended mode
- The problem cannot be resolved through problem analysis
- An error log ID is not available for a problem that might be resolved by using problem analysis
- The LOGPRB parameter for the message is set to *NO

Element 2: Message Data Field Format Number

***NONE:** No message data field format number is passed with the alert identifier.

format-number: Specify the message data field format number that is passed with the alert identifier.

LOGPRB

Specifies, for IBM-supplied messages, whether an entry is put into the problem log. If there is an error log ID for the message and *YES is specified for this parameter, the user can press the F14 key from the system operator message queue display (Display Messages) to call for problem analysis.

***SAME:** The value does not change.

***NO:** The entry is not put in the problem log.

***YES:** The entry is put in the problem log if there is an error log ID associated with the message.

Examples

Example 1: Changing the First-Level Message and Severity

```
CHGMSGD   MSGID(UIN0115)  MSGF(INV)
  MSG('Enter your name') SEV(55)
```

This command changes the first-level message and the severity of message UIN0115 stored in the message file INV. The rest of the message description remains as originally specified in the ADDMSGD command.

Example 2: Changing to a Range of Valid Replies

Assume the user created message UPY0047 as follows:

```
ADDMSGD   MSGID(UPY0047)  MSGF(PAYLIB/TIMECARD)
  MSG('Enter department number:')
  TYPE(*DEC)  LEN(4)  VALUES(0816 0727 0319 8774)
```

To change to a range of valid replies (RANGE parameter), instead of specific reply values (as specified with the VALUE parameter), the following command can be used:

```
CHGMSGD   MSGID(UPY0047)  MSGF(PAYLIB/TIMECARD)
  VALUES(*NONE)  RANGE(0300 8900)
```

The VALUES as originally defined are removed and the RANGE parameters are added to the message description. The type and length of the reply values do not change.

Note: All changes made to an existing message description must be compatible with the existing message description. For example, the following change would be diagnosed as invalid because the RANGE values are not compatible with the reply length as defined on the original ADDMSGD command.

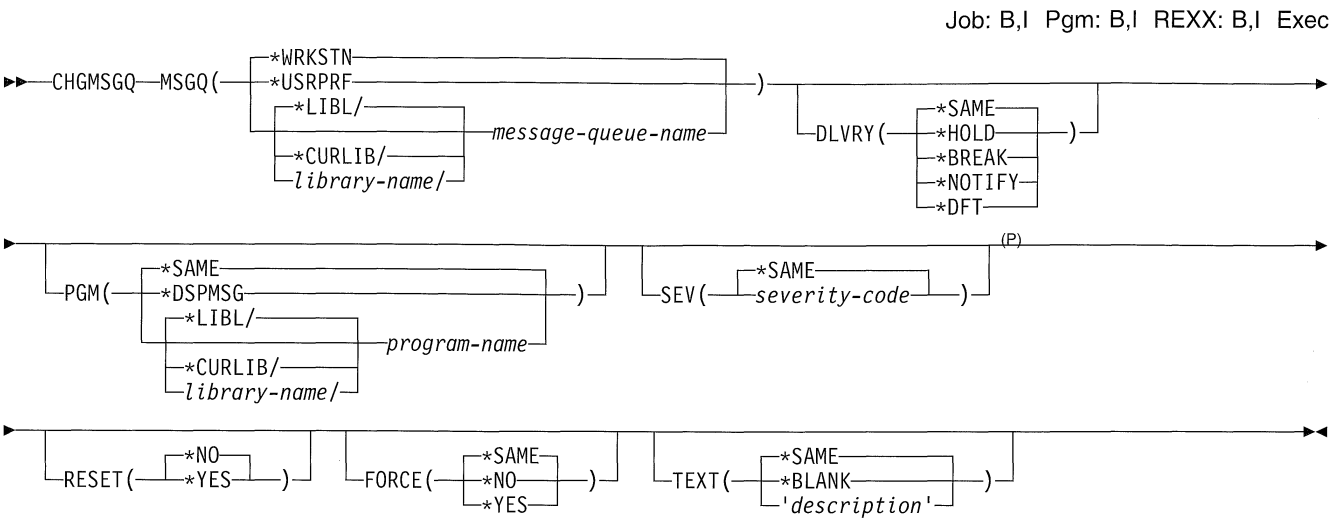
Example 3: Changing the Length Parameter

```
ADDMSGD   MSGID(XYZ0202)  MSGF(XYZMSGF)
  MSG('Enter routing code:')
  TYPE(*CHAR) LEN(2)
  VALUES(AA BB CC DD EE)

CHGMSGD   MSGID(XYZ0202)  MSGF(XYZMSGF)
  VALUES(*NONE) RANGE(AAA ZZZ)
```

To make the change to the range of reply values valid, the user must also change the length (LEN parameter). The correct command coding would be as follows:

```
CHGMSGD   MSGID(XYZ0202)  MSGF(XYZMSGF)
  LEN(3)  VALUES(*NONE) RANGE(AAA ZZZ)
```

CHGMSGQ (Change Message Queue) Command



Note:

P All parameters preceding this point can be specified in positional form.

Purpose

The Change Message Queue (CHGMSGQ) command changes the attributes of the specified message queue. If the delivery mode is being changed to *BREAK or *NOTIFY and if the message queue is not already in *BREAK or *NOTIFY mode or specifically allocated to another job, it is implicitly allocated by this command. The DLVRY, PGM, and SEV parameters are not contained in the Create Message Queue (CRTMSGQ) command, but default values are assigned to them by the system when the message queue is created. This command can also be used to reset the status of old messages to that of new messages so they can be received again without the use of message reference keys.

Restrictions:

1. The user must have *USE and *DELETE authorities for the queue.
2. The message queue being changed cannot be allocated with either a *SHRRD or *EXCL lock by another job. If the queue is allocated by another job when the CHGMSGQ command is attempted, an error message will be sent.

Required Parameter

MSGQ

Specifies the name of the message queue whose attributes are being changed.

***WRKSTN:** The attributes of the work station's message queue are being changed. This parameter is not valid for a batch job.

***USRPRF:** The attributes of the message queue associated with the signed-on user profile are being changed.

The name of the message queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

message-queue-name: Specify the name of the message queue whose attributes are being changed.

Optional Parameters

DLVRY

Specifies how the messages sent to this message queue are delivered. The method of delivery is in effect only as long as the message queue is allocated to the job. When the queue is deallocated, the delivery mode is changed to *HOLD for work station, system operator, and user message queues. However, if *DFT is specified for a system operator or user message queue, the delivery mode remains *DFT.

Note: When the delivery mode of a message queue is changed to *BREAK or *NOTIFY, the message queue is allocated with an *EXCL lock. The *EXCL lock will remain on the queue until the delivery mode is changed to *HOLD or *DFT, or the job that contains the lock on the queue ends.

***SAME:** The value does not change.

***HOLD:** The messages are held in the message queue until they are requested by the user or program. The display station user uses the Display Messages (DSPMSG) command to show the messages; a program must issue a Receive Message (RCVMSG) command to receive a message and handle it.

***BREAK:** The job to which the message queue is allocated is interrupted when a message arrives at the message queue, and the program specified in the PGM parameter is called. If the job is an interactive job, the audible alarm is sounded (if installed). The delivery mode cannot be changed to *BREAK if the message queue is also being used by another job.

When changing to *BREAK mode, the PGM parameter defaults to PGM(*DSPMSG) if a program name is not specified or if PGM(*SAME) is specified.

***NOTIFY:** The job to which the message queue is allocated is notified when a message arrives at the message queue. For interactive jobs at a display station, the audible alarm is sounded and the Message Waiting light is turned on. For batch jobs, no notification occurs; the message is simply held in the queue (the same as for *HOLD). The delivery mode cannot be changed to *NOTIFY if the message queue is also being used by another job.

***DFT:** The default reply to the inquiry message is sent. If no default reply is specified in the message description of the inquiry message, the system default reply, *N, is used. The message is not put on the message queue unless the message queue is QSYSOPR.

PGM

Specifies the qualified name of the program called when a message arrives at the message queue and break delivery has been specified. Because the QSYSOPR message queue receives messages that require manual operator action, only *DSPMSG should be specified or assumed if the message queue being changed is QSYSOPR. The following parameters are passed to the break delivery program:

- Message queue name (10 characters). The name of the message queue to which the message was sent.
- Library name (10 characters). The name of the library containing the message queue.
- Message reference key (4 characters). The reference key of the message sent to the message queue.

***SAME:** The value does not change, unless the message queue is being changed to break delivery. Then the value is changed to *DSPMSG.

***DSPMSG:** The DSPMSG command is processed when a message arrives for break delivery. For interactive jobs, the messages are shown on the display. Also, at the display station, the audible alarm is sounded (if these are installed) and the Message Waiting light is turned on. For batch jobs, the message is sent to a spooled printer file.

The name of the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the program that is called when a message arrives for break delivery.

SEV

Specifies the lowest severity code that a message can have and still be delivered to a user in break or notify mode. Messages arriving at the message queue whose severities are lower than the severity code specified on this parameter do not interrupt the job or turn on the audible alarm or the message-waiting light; they are held in the queue until they are requested by using the Display Message (DSPMSG) command. If *BREAK or *NOTIFY is specified on the DLVRY parameter, and is in effect when a message arrives at the queue, the message is delivered if the severity code associated with the message is equal to or greater than the value specified here. Otherwise, the message is held in the queue until it is requested.

***SAME:** The value does not change.

severity-code: Specify severity code, ranging from 00 through 99.

RESET

Specifies whether old messages (messages that have been received once and were not removed from the message queue) held in the message queue are reset to the new message status. The messages can then be received in first-in, first-out (FIFO) order as they were originally. This parameter applies only to messages received by a program; it does not affect message displays. If all messages are being cleared, refer to the RMVMSG (Remove Message) command description.

***NO:** Old messages in the message queue are not reset to new message status. To receive an old message, to reply to it, or to remove it, enter the message reference key.

***YES:** All messages in the message queue, except inquiry messages that have been sent a reply, are reset to the new message status. These messages can then be received as new messages in the same order that they were sent to the message queue.

FORCE

Specifies whether changes made to the message queue description or messages added to or removed from the queue are immediately forced into auxiliary storage.

This ensures that changes to the queue, or messages sent or received, are not lost if a system failure occurs.

***SAME:** The value does not change.

***NO:** Changes made to the message queue, including its messages, are not immediately forced to auxiliary storage.

***YES:** All changes to the message queue description and to the messages in the queue are immediately forced to auxiliary storage. This can result in program performance problems.

TEXT

Specifies text that briefly describes the message queue. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Examples

Example 1: Changing Method of Delivery to Notify Mode

```
CHGMSGQ   MSGQ(JONES)  DLVRY(*NOTIFY)
```

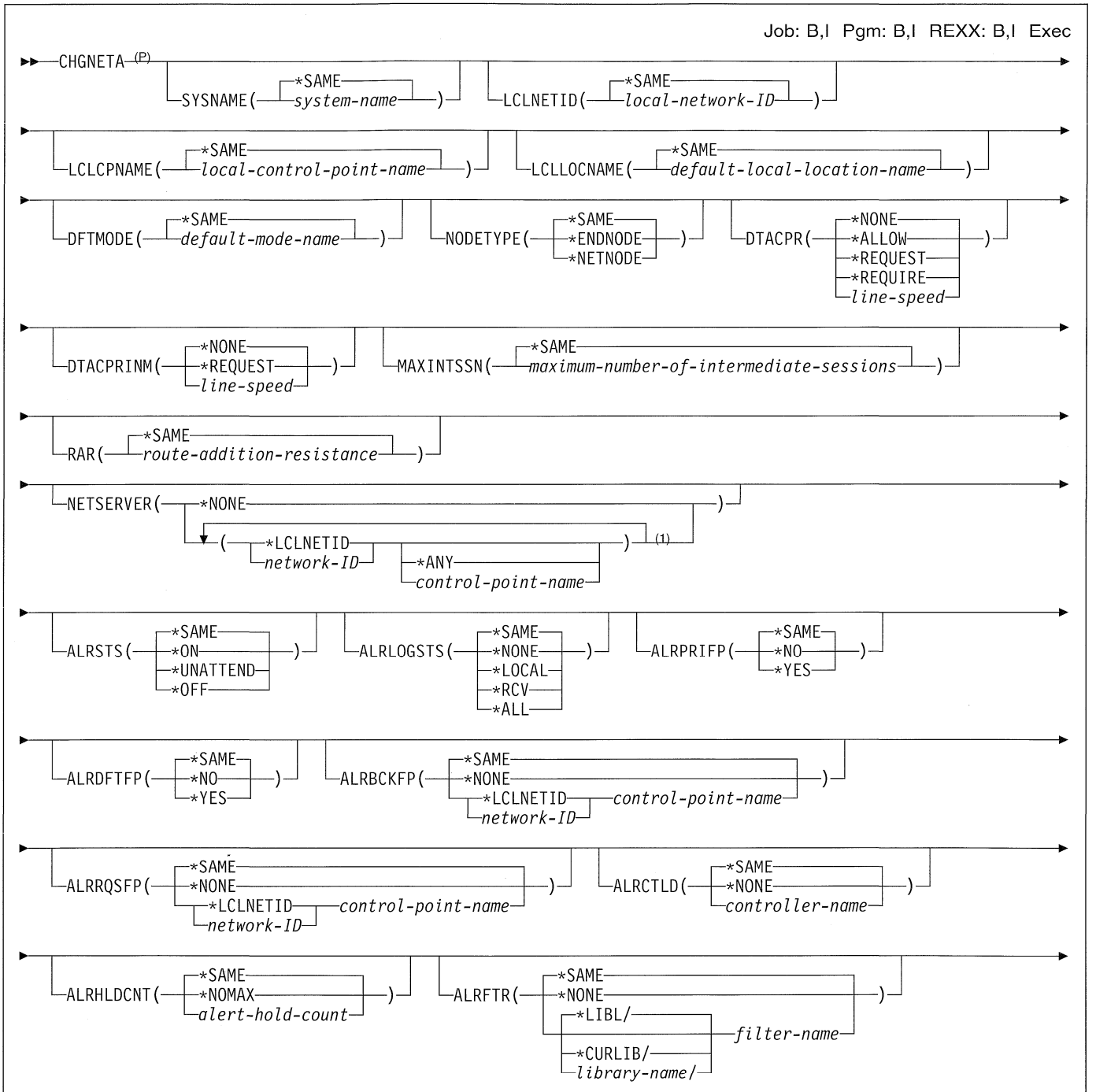
This command changes the method of delivery of the message queue named JONES to notify mode. The user is immediately notified by the attention light and audible alarm (if installed) when a message has been sent to his queue.

Example 2: Changing Method of Delivery to Break Mode

```
CHGMSGQ   MSGQ(INV)  DLVRY(*BREAK)  PGM(INVUPDT)
```

This command changes the delivery mode of the message queue named INV to *BREAK and calls a program named INVUPDT when a message arrives at INV.

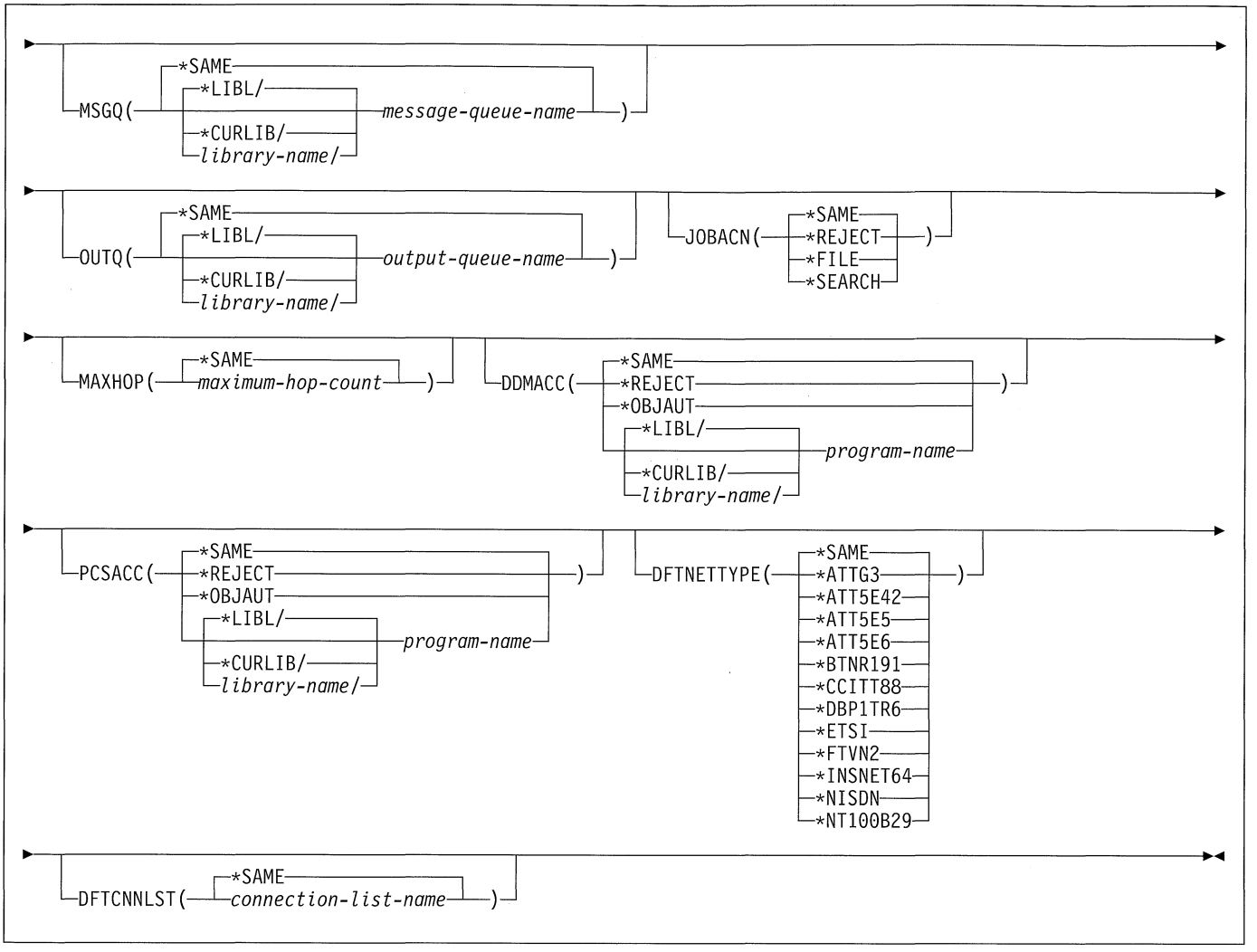
CHGNETA (Change Network Attributes) Command



Notes:

P All parameters preceding this point can be specified in positional form.

1 A maximum of 5 repetitions



Purpose

The Change Network Attributes (CHGNETA) command changes the network attributes of a system.

Network attributes are saved by the Save System (SAVSYS) command: network attributes are optionally restored to the system when the operating system is installed. The following network attributes are not restored if they are saved from another system: system name, local network ID, local control point name, default local location name, default mode, node type, maximum number of intermediate sessions, route addition resistance, network node servers, alert primary focal point, and alert default focal point.

When a new default local location name, control point name, or network ID is specified, a check is made against the remote location list to determine whether the new qualified local location name or control point name already exists on the list. If it does, an error occurs and no network attributes change.

If there are synchronous datalink control (SDLC) line descriptions that are varied on and the short hold mode node type (SHMNODE) parameter has a value of *T21, the physical interface (INTERFACE) parameter has a value of *X21, and the connection type (CNN) parameter has a value of *SHM, an attempt to change the network ID and the control point name causes an error, and no network attributes change.

If any APPC controllers with the APPN option set to *YES are varied on, an attempt to change the local control point name, the network ID, or the node type causes an error, and no network attributes change.

If a new default local location name or network ID is entered, devices are checked to ensure that descriptions are still valid. When a new default local location name is entered, the entry is checked against existing device descriptions with LCLLOCNAME(*NETATR) to ensure that the LCLLOCNAME/RMTLOCNAME pair has different values. If either the default local location or network ID is changed, all devices attached to the same controller with LCLLOCNAME(*NETATR) or RMTNETID(*NETATR) are checked to ensure that each device description remains unique. If an error occurs, no network attributes change.

A user is required to have both *ALLOBJ and *SECADM authority to change the network attributes JOBACN, DDMACC, and PCSACC.

Restriction: This command is shipped with public *EXCLUDE authority. When this command is shipped, authority is issued only to the security officer. The security officer can grant the use of this command to other users.

More information on this command is in the *Work Management Guide* or the *Communications Management Guide*.

Optional Parameters

SYSNAME

Specifies the name assigned to the system. The name can contain up to 8 alphanumeric characters. The characters that are allowed are uppercase letters A through Z, 0 through 9, or special characters @, #, $, and embedded blanks. Embedded blanks must be enclosed in single quotation marks (') in the same manner as other character strings. Leading blanks are not allowed.

If the system name is not set, it is defaulted at initial program load (IPL) to the machine serial number with the first character changed, if necessary, to an alphabetic character.

Note: The use of the @, #, and $ characters is not recommended because they are not part of an invariant character set and are not available on all keyboards.

***SAME:** The value does not change.

system-name: Specify the name that identifies the system. The pending system name is changed when a request to change the system name is issued. The pending system name becomes the current system name at the next IPL of the system. If the system name is changed during the IPL, the name specified becomes the current system name.

LCLNETID

Specifies the local network identifier (ID) assigned to the system. The name can contain up to 8 alphanumeric characters. The first character must be an uppercase letter A through Z, or special character $, #, or @ and cannot contain blanks, plus signs (+), periods (.), or underscores (_). More information on these requirements is in the *APPC Programmer's Guide*.

***SAME:** The value does not change.

local-network-ID: Specify the ID of the local network. The network ID is changed if all APPC controllers with the APPN parameter set to *YES are varied off.

LCLCPNAME

Specifies the name of the local control point for the system. The name can contain up to 8 alphanumeric characters. The first character must be an uppercase letter A through Z, or special character $, #, or @ and cannot contain blanks, plus signs (+), periods (.), or underscores (_). More information on these requirements is in the *APPC Programmer's Guide*. If the local control point name is not set, it is defaulted at IPL to the machine serial number with the first character changed (if necessary) to an alphabetic character.

***SAME:** The value does not change.

local-control-point-name: Specify the local control point name for the system. The control point name is changed if all APPC controllers with the APPN parameter set to *YES are varied off.

LCLLOCNAME

Specifies the local location name.

Note: The name may contain up to 8 alphanumeric characters. The first character must be an uppercase letter A through Z, or special character $, periods (.), or underscores (_). If the default local location name is not set, it is defaulted at IPL to the machine serial number (with the first character changed, if necessary, to an alphabetic character). More information is in the *APPC Programmer's Guide*.

***SAME:** The value does not change.

default-local-location-name: Specify the default local location name for the system.

DFTMODE

Specifies the default mode name for the system. The name can contain up to 8 alphanumeric characters in length. The first character must be an uppercase letter A through Z, or special character $, #, or @ and cannot contain blanks, plus signs (+), periods (.), or underscores (_). More information is in the *APPC Programmer's Guide*.

***SAME:** The value does not change.

default-mode-name: Specify the default mode name. Mode names SNASVCMG and CPSVCMG are not allowed. Specifying BLANK for the default mode name is the same as specifying a mode name of all blanks.

NODETYPE

Specifies the APPN node type. More information is in the *APPC Programmer's Guide*.

***SAME:** The value does not change.

***ENDNODE:** The node does not provide network services to other nodes, but it can participate in the APPN network by using the services of an attached network server or it can operate in a peer-to-peer environment similar to migration end nodes.
NODETYPE(*ENDNODE) cannot be specified if either ALRPRIFP(*YES) or ALRDFTFP(*YES) is specified.

***NETNODE:** The node provides intermediate routing and route selection services and distributed directory services for local users to end nodes and migration end nodes that it is serving.

DTACPR

Specifies the level of data compression used when the AS/400 system is an SNA end node.

***NONE:** Data compression is not allowed on the session.

***ALLOW:** Data compression is allowed on the session by the local system if requested by a remote system. The local system does not request compression.

If data compression is requested by the remote system, the data compression levels used by the session are the lower of the requested levels and the configured levels.

***REQUEST:** Data compression is requested on the session by the local system. However, the request can be refused or changed to a lower compression level by the remote system. Data compression is allowed on the session if requested by the remote system.

The data compression levels that the local system requests are the configured levels. If data compression levels are changed by the remote system, they cannot exceed the configured values.

If data compression is requested by the remote system, the data compression levels used by the session are the lower of the requested levels and the configured levels.

***REQUIRE:** Data compression is required on the session. If the remote system does not accept the local system's exact required levels of compression, the session is not established.

The data compression levels that the local system requires are the configured levels.

line-speed: Specify the maximum line speed at which data is compressed. If the line speed of the link used by the session is less than or equal to this specified line speed, data compression is used for the session as if *REQUEST is specified. Otherwise, compression is used for the session as if *ALLOW is specified. Valid values range from 1 through 2147483647 in bits per second (bps).

DTACPRINM

Specifies the level of data compression to request when the AS/400 system is an SNA intermediate node.

***NONE:** The remote systems are not notified of a need to compress data when the AS/400 system is an SNA intermediate node.

***REQUEST:** The remote systems are requested to compress data when the AS/400 is an SNA intermediate node.

line-speed: Specify a line speed. If either the receiving or sending link has a line speed equal to or less than this specified line speed, data compression is requested from the remote systems. Otherwise, remote systems are not requested to compress the data. Valid values range from 1 through 2147483647 bits per second (bps).

MAXINTSSN

Specifies the maximum number of APPN intermediate sessions. This parameter is used only if NODETYPE(*NETNODE) is specified. More information is in the *APPC Programmer's Guide*.

***SAME:** The value does not change.

maximum-number-of-intermediate-sessions: Specify the maximum number of intermediate sessions. Valid values range from 0 through 2000.

RAR

Specifies the APPN route addition resistance (RAR). This parameter is used only if NODETYPE(*NETNODE) is specified. More information is in the *APPC Programmer's Guide*.

***SAME:** The value does not change.

route-addition-resistance: Specify the RAR. Valid values range from 0 through 255.

NETSERVER

Specifies the APPN network node servers. This parameter is used only if *ENDNODE is specified on the NODETYPE parameter. More information is in the *APPC Programmer's Guide*.

A network node server is a network identifier (ID) qualified by a control point name. Up to five network node servers can be specified. A network node server must be specified by using two values (one value for each element): the network ID and the control point name. Name restrictions are the same as those for LCLNETID and LCLCPNAME.

***NONE:** The list of network node servers is cleared. This value can be specified as the only value for NETSERVER.

Element 1: Server Network ID

***LCLNETID:** The value used depends on the value of the network ID network attribute (LCLNETID) at the time the server is referenced.

network-ID: Specify the local network ID that becomes the network node server.

Element 2: Control Point Name

***ANY:** The first network node that offers services becomes the network node server. Any network node with the same network ID as that specified on the LCLNETID parameter can be a network node server.

control-point-name: Specify the control point name of the network node server.

ALRSTS

Specifies whether local alerts are generated. When alert processing is active, alerts are generated when a permanent error is encountered with local devices or control units, network lines, devices, and control units, and when programming and operator errors occur. More information is in the *Alerts and DSNX Guide* and the *APPC Programmer's Guide*.

***SAME:** The value does not change.

***ON:** Alert processing is started. The system generates alerts for all alert conditions except unattended conditions.

***UNATTEND:** The system generates alerts for all alert conditions including those that have the alert indicator in the message description set *UNATTEND.

***OFF:** No monitoring of errors occurs.

ALRLOGSTS

Specifies how alerts are logged by the AS/400 system. More information on the logging of system alerts is in the *Alerts and DSNX Guide*.

***SAME:** The value does not change.

***NONE:** No alerts are logged.

***LOCAL:** Only locally generated alerts are logged.

***RCV:** Only alerts received from other systems are logged.

***ALL:** Both locally generated alerts and alerts received from other systems are logged.

ALRPRIFP

Specifies whether the system is an alert primary focal point. If the system is defined as a primary focal point, alerts are received from all network nodes explicitly defined in the sphere of control. More information is in the *Alerts and DSNX Guide* and the *APPC Programmer's Guide*.

***SAME:** The value does not change.

***NO:** The system is not an alert primary focal point. Information about when this parameter can be changed from *YES to *NO is in the *Alerts and DSNX Guide*. Use the Display Sphere of Control Status (DSPSOCSTS) command to see the status for all systems in the sphere of control.

***YES:** The system is defined as an alert primary focal point and it provides focal point services to all systems in the network that are explicitly defined in the sphere of control. If a system is defined as a focal point, ALRLOGSTS(*ALL) or ALRLOGSTS(*RCV) must be specified to ensure that alerts coming in from nodes in the sphere of control are logged.

ALRDFTFP

Specifies whether the system is a default alert focal point. If the system is defined as a default alert focal point, alerts are received from all network systems not explicitly defined in the sphere of control of some other focal point system within the network. More information is in the *Alerts and DSNX Guide*.

***SAME:** The value does not change.

***NO:** The system is not a default alert focal point. Information about when this parameter can be changed from *YES to *NO is in the *Alerts and DSNX Guide*. Use the Display Sphere of Control Status (DSPSOCSTS) command to show the status for all systems in the sphere of control.

***YES:** The system is a default alert focal point and it provides focal point services to all network systems not being serviced by either a primary focal point or another default focal point. If a system is defined as a focal point, NODETYPE(*NETNODE) must be specified.

ALRBCKFP

Specifies the name of the system that provides alert focal point services if the primary focal point is unavailable.

***SAME:** The value does not change.

***NONE:** The backup focal point is not defined.

Element 1: Network ID

***LCLNETID:** The network ID of the backup focal point is same as that of the local system.

network-ID: Specify the network ID of the system that provides backup focal point services for alerts.

Element 2: Control Point Name

control-point-name: Specify the control point name of the system that provides backup focal point services for alerts.

ALRRQSFP

Specifies the name of the system that is requested to provide alert focal point services.

***SAME:** The value does not change.

***NONE:** A focal point is not requested. The current focal point is revoked.

Element 1: Network ID

***LCLNETID:** The network ID of the requested focal point is same as that of the local system.

network-ID: Specify the network ID of the system that is requested to provide focal point services for alerts.

Element 2: Control Point Name

control-point-name: Specify the control point name of the system that is requested to provide focal point services for alerts.

ALRCTLD

Specifies the name of the controller through which alerts are sent on the alert controller session. Only a host or APPC controller can be specified. The controller must be varied on for alert processing to be operational on the alert controller session, although it does not need to be varied on when this command is used. More information about the alert controller session is in the *Alerts and DSNX Guide*.

***SAME:** The value does not change.

***NONE:** No alert controller is described.

controller-name: Specify the name of the controller being used for alerts in an alert controller session. This controller is ignored if the system has a primary or

default alert focal point (if, for example, the node is in another system's sphere of control).

ALRHLDCNT

Specifies the maximum number of alerts that are created before the alerts are sent over the alert controller session (ALRCTLD network attribute). The alerts are held (queued) by the system until the specified number of alerts have been created. This parameter can be used to manage alerts that are sent over a limited resource by reducing the number of times alerts are sent.

Note: The ALRHLDCNT is valid only when the ALRCTLD parameter is specified. When management services sessions, APPN, and sphere of control support are used, the ALRHLDCNT parameter is ignored.

***SAME:** The value does not change.

***NOMAX:** The alerts are held indefinitely. The current alert hold count is the maximum value. The alerts can be sent at a later time by changing the ALRHLDCNT value to a lower value.

alert-hold-count: Specify the maximum number of alerts that can be created before being sent. Alerts have a "held" status until the maximum is reached. If the value 0 is specified, alerts are sent as soon as they are created. Valid values range from 0 through 32767.

ALRFTR

Specifies the qualified name of the alert filter used by the alert manager when processing alerts.

***SAME:** The value does not change.

***NONE:** An alert filter is not used.

The name of the alert filter can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

filter-name: Specify the name of the alert filter.

MSGQ

Specifies the qualified name of the message queue where messages received through the SNADS network are sent for users with no message queue specified in their user profile, or whose message queue is not available.

***SAME:** The value does not change.

The possible library values are:

***LIBL:** The library list is used to locate the message queue.

When *LIBL is used as the library name, the library list of the job calling this command is searched to find a message queue with the specified object name. If the message queue is found, the name of the library in which it is found is used in the fully qualified name and it is stored. If the message queue is not found, an exception is signaled and no network attributes are changed.

***CURLIB:** The current library for the job is used to locate the message queue. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the message queue is located.

When the library name or *CURLIB is specified, this command attempts to find the message queue. If the message queue cannot be found in the specified library, a diagnostic message is sent. If all other parameters on the command are specified correctly, and whether or not this command is able to find the message queue in the library specified, the MSGQ network attribute is changed to the qualified message queue name.

message-queue-name: Specify the name of the message queue.

OUTQ

Specifies the qualified name of the output queue.

***SAME:** The value does not change.

The possible library values are:

***LIBL:** The library list is used to locate the output queue.

When *LIBL is used as the library name, the library list of the job calling this command is searched to find an output queue with the specified object name. If the output queue is found, the name of the library in which it is found is used in the qualified name and it is stored. If the output queue is not found, an exception is signaled, and no network attributes are changed.

***CURLIB:** The current library for the job is used to locate the output queue. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the output queue is located.

When the library name or *CURLIB is specified, this command attempts to find the output queue. If the output queue cannot be found in the specified library, a diagnostic message is sent. If all other parameters on the command were specified correctly, and whether or not this command finds the output queue in the specified library, the OUTQ network attribute is changed to the output queue name.

output-queue-name: Specify the name of the output queue.

JOBACN
Specifies the action taken for job input streams received through the SNADS network by the system.

***SAME:** The value does not change.

***REJECT:** The input stream is rejected by the system. This action allows the user to secure the system from input streams received through the network.

***FILE:** The input stream is filed in the queue of network files received for the user to which it was sent. That user can then look at, end, or receive the input stream, or submit it to a job queue.

***SEARCH:** The table of network job entries is searched to determine the action taken for the input stream.

MAXHOP
Specifies the maximum number of times in a SNADS network that a distribution queue entry originating at this node can be received and routed on the path to its final destination. If this number is exceeded, the distribution queue entry is canceled, and a feedback status is sent to the sender.

***SAME:** The value does not change.

maximum-hop-count: Specify the maximum number of times that a distribution queue entry can be received and routed before it is canceled. Valid values range from 1 through 255.

DDMACC
Specifies how the system processes distributed data management (DDM) requests from remote systems for access to the data resources of the system. You must specify a special value or program name which dictates how the requests are to be handled. Changes to this parameter are immediate. However, jobs currently running on the system do not use the new value. The DDMACC value is used only when a job is first started.

***SAME:** The value does not change.

***REJECT:** DDM requests from remote systems are not processed. However, this system can still use DDM access files on remote systems. Source systems cannot access files on an AS/400 system that specifies *REJECT.

***OBJAUT:** All file requests are accepted if the user profile associated with the DDM job is authorized to the files. Object authorities such as read, write, or update, must also exist for the files.

The possible library values are:

***LIBL:** The library list is used to locate the program.

If *LIBL is used as the library name, the library list of the job calling this command is searched to find the program name with the specified object name. If the program name is found, the name of the library in which it is found is used in the fully qualified name and it is stored. If the program name is not found, an error message is sent and no network attributes are changed.

***CURLIB:** The current library for the job is used to locate the program. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the program is located.

When the library name or *CURLIB is specified, this command attempts to find the program name. If the program name cannot be found in the specified library, a diagnostic message is sent to the user. If this command is able to find the program name in the specified library, the DDMACC network attribute is changed to the qualified program name, if all other parameters on the command are specified correctly.

program-name: Specify the name of the program.

The program name is the name of the customer validation program that can supplement system object level security. The user security exit program parameter list is the same parameter list used by the personal computer Support/38 applications. This user-exit program can restrict user access to *PUBLIC and privately authorized files. The target DDM support calls the user program each time a file is read. The user exit program indicates to DDM whether the request must proceed or end. AS/400 system object level security still applies. More information about this program is in the *Data Management Guide*.

PCSACC
Specifies how PC Support requests are handled. You must specify a special value or program name which dictates how the requests are to be handled. This permits greater control over PC Support applications. Changes to this parameter are immediate. However, jobs currently running on the system do not use the new value. The PCSACC value is used only when a job is first started.

***SAME:** The value does not change.

***REJECT:** The system rejects all requests from the personal computer.

***OBJAUT:** Normal object authorizations are checked for this personal computer request (for example, authorization to retrieve data from a database file for a transfer facility request).

The possible library values are:

***LIBL:** The library list is used to locate the program.

When *LIBL is used as the library name, the library list of the job calling this command is searched to find the program name with the specified object name. If the program name is found, the name of

the library in which it is found is used in the fully qualified name and it is stored. If the program name is not found, an error message is sent and no network attributes are changed.

***CURLIB:** The current library for the job is used to locate the program. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the program is located.

When the library name or *CURLIB is specified, this command attempts to find the program name. If the program name cannot be found in the specified library, a diagnostic message is sent to the user. If all other parameters on the command were specified correctly, and whether or not this command is able to find the program name in the specified library, the PCSACC network attribute is changed to the qualified program name.

program-name: Specify the name of the program.

The program name is the name of the customer-supplied PC Support host system application exit program that can supplement system object level security. This user-exit program can restrict requests handled from the personal computer. Each personal computer support application calls the exit program for requests from the personal computer. Two parameters are passed to the user-exit program: the first describes the personal computer request (which application and what kind of request). The second is used by the exit program to indicate to the personal computer support application whether this personal computer request must be handled. More information about this program is in the *PC Support/400 Technical Reference for DOS and OS/2*.

DFTNETTYPE

Specifies the system default value for the integrated services digital network (ISDN) network type.

This parameter affects the creation of network interface and connection list objects on the system. The value specified here becomes the default network type (*NETTYPE) for connection lists and network interface descriptions created using the Create Connection List (CRTCNNL) command and the Create Network Interface description for ISDN (CRTNWIISDN) command.

***SAME:** The value does not change.

***ATTG3:** This value is used when attaching to an ISDN in the United States or Canada that uses AT&T Definity G3i switching equipment.

***ATT5E42:** This value is used when attaching to an ISDN in the United States or Canada that uses AT&T 5ESS version 5E4.2 switching equipment.

***ATT5E5:** This value is used when attaching to an ISDN in the United States or Canada that uses AT&T 5ESS version 5E5 switching equipment.

***ATT5E6:** This value is used when attaching to an ISDN in the United States or Canada that uses AT&T 5ESS version 5E6 switching equipment.

***BTNR191:** This value is used when attaching to an ISDN controlled by British Telecomm in the United Kingdom.

***CCITT88:** This value is used when attaching to an ISDN that follows the 1988 recommendations of the International Telegraph and Telephone Consultative Committee (CCITT).

***DBP1TR6:** This value is used when attaching to an ISDN controlled by Germany's post telephone and telegraph administration (PTT).

| ***ETSI:** This value is used when attaching to an ISDN
| that uses the European Telecommunications Standards
| Institute (ETSI, also known as EuroISDN) standard.

***FTVN2:** This value is used when attaching to version 2 of the ISDN controlled by France's PTT.

***INSNET64:** This value is used when attaching to the INSNET64 ISDN controlled by Japan's Nippon Telephone and Telegraph Public Corporation (NTT).

| ***NISDN:** This value is used when attaching to an ISDN
| that uses the National ISDN-1 or National ISDN-2
| standard for North America.

***NT100B29:** This value is used when attaching to an ISDN in the United States or Canada that uses Northern Telecom DMS100 Version BCS-29 switching equipment.

DFTCNNLST

Specifies the system default value for the ISDN connection list. The connection list contains information for identifying authorized incoming calls.

This parameter affects the creation of ISDN Data Link Control (IDLC) line description objects on the system. The value specified here becomes the default connection list for line descriptions created using the Create Line Description for IDLC (CRTLINIDLC) command.

***SAME:** The value does not change.

connection-list-name: Specify the name of the default connection list.

Examples

Example 1: Changing the Pending System Name

```
CHGNETA  SYSNAME(SYSTEST)  ALRSTS(*ON)  ALRPRIFP(*YES)
  ALRLOGSTS(*LOCAL)
```

This command changes the pending system name. The current system name is changed at the next IPL. Local alerts are generated and logged and the system is a primary focal point.

Example 2: Changing the Node Servers

```
CHGNETA  LCLCPNAME(CPNAME)  LCLNETID(NETNAME)
  NETSERVER((*LCLNETID  BOSTON)
  (MINN ROCHEST)  (MAINE BANGOR))
```

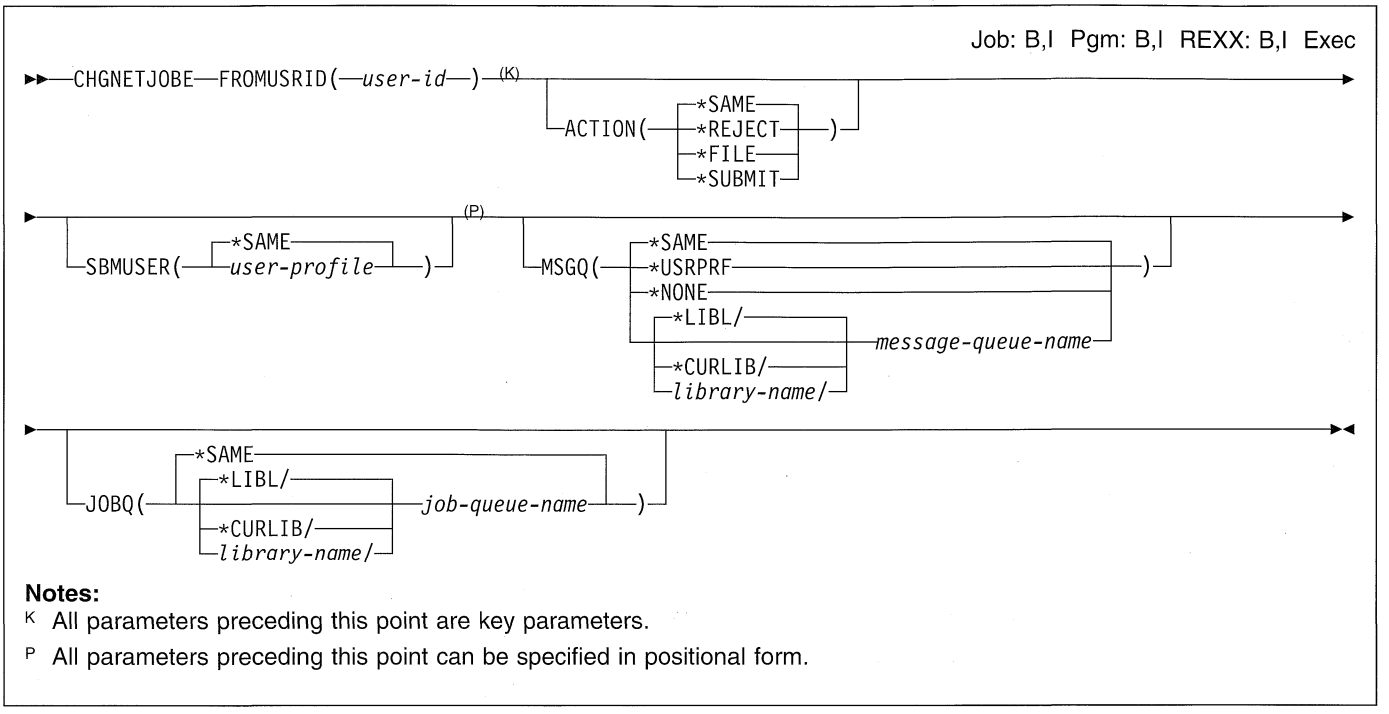
This command changes the name of the local control point, the network ID, and the list of network node servers. The servers are used only if the node type is *ENDNODE.

Example 3: Clearing the List of Node Servers

```
CHGNETA  NETSERVER(*NONE)  DDMACC(DDMLIB/DDMPGM)
```

This command clears the list of network node servers and sets the name of the DDM access program.

CHGNETJOBE (Change Network Job Entry) Command



Notes:

K All parameters preceding this point are key parameters.

P All parameters preceding this point can be specified in positional form.

Purpose

The Change Network Job Entry (CHGNETJOBE) command changes an existing network job entry in the system. The network job entry is used to determine the action taken when an input stream is sent to a user by using the Submit Network Job (SBMNETJOB) command. This entry determines whether the input stream is automatically submitted, placed on the queue of network files for a user, or rejected. The entry also specifies the user profile that is used for checking the authority to the job description referenced in the input stream. There should be one entry for each user or distribution group who submits jobs to the system.

Note: There is a network attribute, Job Action (JOBACN), that provides overall control of network job submission; its value must be *SEARCH before the network job table is searched for an action. If the network attribute is *REJECT, all incoming jobs are rejected. If the network attribute is *FILE, all incoming network jobs are saved in the user's queue of network files regardless of the network job entry.

Each network job entry is identified by the two-part user ID of the sender. When an input stream arrives, the user ID of the sending user is used to find a network job entry. If no entry is found, the second part of the user ID is used to find an entry, using *ANY for the first part. If this search fails, a search is made using *ANY for both parts of the user ID. If no entry is found, the job is rejected.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. The internal value for a node identifier may differ from the characters shown by the CHGNETJOBE command depending on the type of work station (language) you are using. If the byte-string value specified for the FROMUSRID parameter does not match the rules for an internal node identifier value, or if it does not match the internal value for any defined node (ignoring case differences), an error may be reported.

Required Parameter

FROMUSRID

Specifies the two-part user ID of the user who submits an input stream to this system. Any input streams received from the user are handled as specified in this network job entry. Both parts of the user ID are required. A special value of *ANY can be entered for the first part or for both parts of the user ID.

Note: Depending on the type of work station you are using, the internal value for a user identifier may differ from the characters shown by the Work with Network Job Entry (WRKNETJOBE) command. If the byte-string value specified for the FROMUSRID parameter does not match the rules for an internal user identifier value, or if it does not match the internal value for any enrolled user, an error may be reported.

Optional Parameters

ACTION

Specifies the action that is taken for a job controlled by this entry.

***SAME:** The value does not change.

***REJECT:** The input stream is rejected.

***FILE:** The input stream is placed on the queue of network files received for the user to whom the input stream is sent.

***SUBMIT:** The input stream is submitted to a batch job queue. The user profile specified in the network job entry is used to check for the required authority to the job queues that are used and to the job descriptions specified in the input stream.

SBMUSER

Specifies the user profile name under which the jobs are submitted. This user profile is used to check the authority to the job queues and job descriptions specified in the input stream. The value specified for this parameter is effective if ACTION(*SUBMIT) is specified either on this command or on the Add Network Job Entry (ADDNETJOBE) command.

***SAME:** The value does not change.

user-profile: Specify the name of the user profile that is used to submit the jobs.

MSGQ

Specifies the qualified name of the message queue to which messages are sent.

Note: The message sent to the message queue specified here notifies the recipient that the input stream arrived, and whether it was submitted, placed on the user's queue of network files, or rejected. A message is also sent to the history log (QHST) when an input stream is received.

***SAME:** The value does not change.

***USRPRF:** The message queue of the user profile to whom the job was sent is used. This user is specified as SBMNETJOB(TOUSRID); this may or may not be the same user as is specified on the SBMUSER parameter of this command.

***NONE:** No message is sent to a user; however, a message is sent to the history log.

The name of the message queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

message-queue-name: Specify the name of the message queue that is used to receive messages. If no library qualifier is specified, the library list (*LIBL) is used to locate the named message queue; if one is found, the name of the library where the message queue was found is placed in the job entry.

JOBQ

Specifies the job queue on which the job entries are placed. A job entry is placed on this queue for each job in the input stream that has JOBQ(*RDR) specified on the Batch Job (BCHJOB) command. If *RDR is not specified on the BCHJOB command, the job queue specified on the BCHJOB command or in the job description is used. (The job queue for each job in the input stream can be different.) This parameter is valid only if ACTION(*SUBMIT) is specified on this command, in the existing network job entry, or in a subsequent Change Network Job Entry (CHGNETJOBE) command.

***SAME:** The job queue in the job entry does not change.

The name of the job queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

If no library qualifier is specified, the library list (*LIBL) is used to locate the named job queue; if one is found, the name of the library where it was found is placed in the job entry.

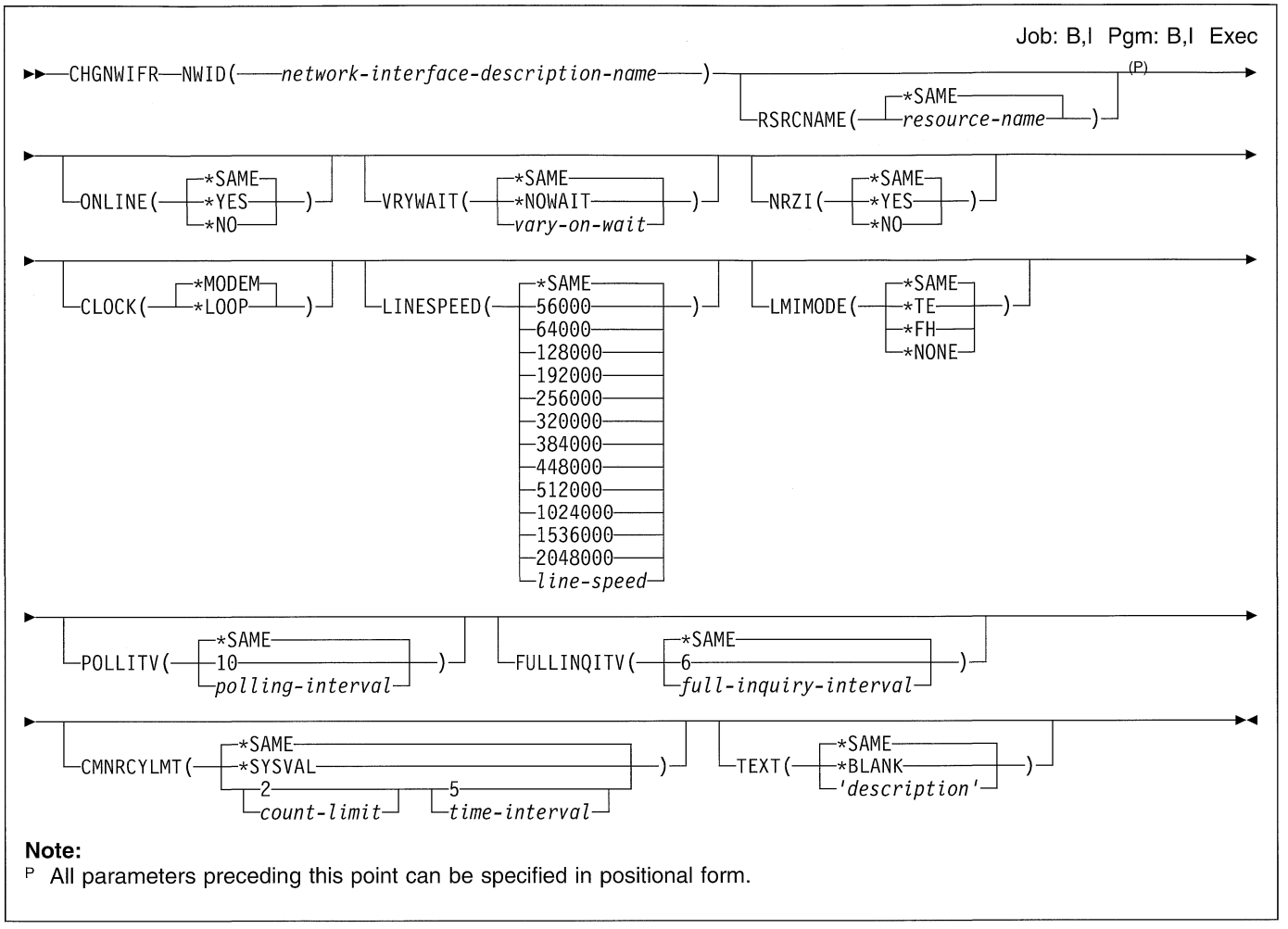
job-queue-name: Specify the name of the job queue.

Example

```
CHGNETJOBE   FROMUSRID(JOHN SMITH)  SBMUSER(ANDERSON)
```

This command changes the network job entry that is used to determine the action taken for any input stream received from user JOHN SMITH. The option and the message queue in the existing entry are not changed. The user profile that is used to check the authority to the job queues and job descriptions specified in the input stream is changed to ANDERSON.

CHGNWIFR (Change Network Interface (Frame Relay Network)) Command



Note:
P All parameters preceding this point can be specified in positional form.

Purpose

The Change Network Interface (Frame-Relay Network) (CHGNWIFR) command changes a network interface for a frame-relay network (FR) interface. More information about using this command is in the *OS/400* Communications Configuration Reference*.

Required Parameter

NWID

Specifies the name of the network interface description.

network-interface-description-name: Specify the name of a network interface description.

Optional Parameters

RSRCNAME

Specifies the resource name that identifies the hardware that the description represents.

Note: Use the Work with Hardware Resources (WRKHDWRSC) command with *CMN specified for the TYPE parameter to help determine the resource name. Specify the resource name of the communications port. The resource name consists of the input/output adapter (IOA) resource name and the port number on the IOA. For example, if the resource name of the IOA is LIN01 and the port on the IOA is 1, then the resource name is LIN011.

***SAME:** The value does not change.

resource-name: Specify a resource name.

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The network interface is automatically varied on at IPL.

***NO:** This network interface is not automatically varied on at IPL.

VRYWAIT

Specifies whether the network interface is varied on asynchronously or synchronously. For synchronous vary on, specifies how long the system waits for the vary on to complete.

***SAME:** The value does not change.

***NOWAIT:** The system does not wait for the vary on to complete. The network interface is varied on asynchronously.

vary-on-wait: Specify the time (in seconds) to wait. Valid values range from 15 through 180. The system waits until the network interface is varied on, or until the specified time passes, before completing the Vary Configuration (VRYCFG) command.

Notes:

1. Specifying a wait time in the network interface description affects system IPL time, if ONLINE(*YES) is used, by the amount of time it takes to synchronously vary on the network interface or reach the wait-time value.
2. The time required to vary on a network interface is the time it takes to put tasks in place to manage the network interface, to activate the communications I/O processor (IOP) (including downloading the IOP model-unique Licensed Internal Code), and to establish communications with the data circuit-terminating equipment (DCE). Normal vary-on time ranges from 5 through 45 seconds, but can be longer, depending on the system, network interface protocol, and other factors.

NRZI

Specifies whether non-return-to-zero-inverted (NRZI) data encoding is used for modems that are sensitive to certain bit patterns in the data stream. This ensures that the signal does not stay the same for an extended period of time.

Note: All data communications equipment on the line must use the same transmission method.

***SAME:** The value does not change.

***YES:** NRZI data encoding is used.

***NO:** NRZI data encoding is not used.

CLOCK

Specifies the method in which the clocking function is provided for the network interface.

***MODEM:** The modem provides the clocking.

***LOOP:** The system inverts the clock from the modem and uses it as the transmit clock on the line.

Note: CLOCK(*LOOP) is not valid when INTERFACE(*RS449V36) or INTERFACE(*V35) is specified.

LINESPEED

Specifies the line speed in bits per second (bps)

***SAME:** The value does not change.

56000: The line speed is 56000 bps.

64000: The line speed is 64000 bps.

128000: The line speed is 128000 bps.

192000: The line speed is 192000 bps.

256000: The line speed is 256000 bps.

320000: The line speed is 320000 bps.

384000: The line speed is 384000 bps.

448000: The line speed is 448000 bps.

512000: The line speed is 512000 bps.

1024000: The line speed is 1024000 bps.

1536000: The line speed is 1536000 bps.

2048000: The line speed is 2048000 bps.

line-speed: Specify the line speed. Valid values range from 56000 bps through 2048000 bps.

LMIMODE

Specifies whether the local management interface (LMI) for this adapter is configured as terminal equipment or a frame handler.

***SAME:** The value does not change.

***TE:** The local system is configured to interface with a frame relay network as terminal equipment.

***FH:** The local system is configured to interface with another system as a frame handler. In this configuration, the local system is performing as the frame relay network.

***NONE:** The local system is configured to interface with the frame relay network or another system without performing any LMI function.

POLLITV

Specifies the rate of the polling cycle. The polling cycle consists of a status enquiry message and a status message exchange. The status message includes the status of the DLCI.

***SAME:** The value does not change.

10: A polling interval of 10 seconds is used.

polling-interval: Specify the polling interval to be used within a 5 to 30 second range.

FULLINQITV

Specifies the number of polling cycles that occur before a full status inquiry is requested.

***SAME:** The value does not change.

6: A full inquiry interval of 6 polling cycles is used.

full-inquiry-interval: Specify the number of polling cycles for a full status cycle to be requested. Valid values range from 1 through 255.

CMNRCYLMT

Specifies the number of recovery attempts made by the system before an inquiry message is sent to the system operator. Also specifies the time (in minutes) that must elapse before the system sends an inquiry message to the system operator indicating that the recovery attempt count limit is reached.

***SAME:** The value does not change.

***SYSVAL:** The recovery limits specified in the QCMNRCYLMT system value are used.

Element 1: Maximum Recovery Limit

2: Two recovery attempts are made within the interval specified.

count-limit: Specify the number of recovery attempts to be made. Valid values range from 0 through 99.

Element 2: Recovery Time Interval

5: The specified number of recovery attempts is made within a 5-minute interval.

time-interval: Specify the time interval (in minutes) at which the specified number of second-level recoveries are attempted. Valid values range from 0 through 120.

If the value specified for *count-limit* is not 0, the value 0 specifies infinite recovery.

TEXT

Specifies text that briefly describes the line description. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

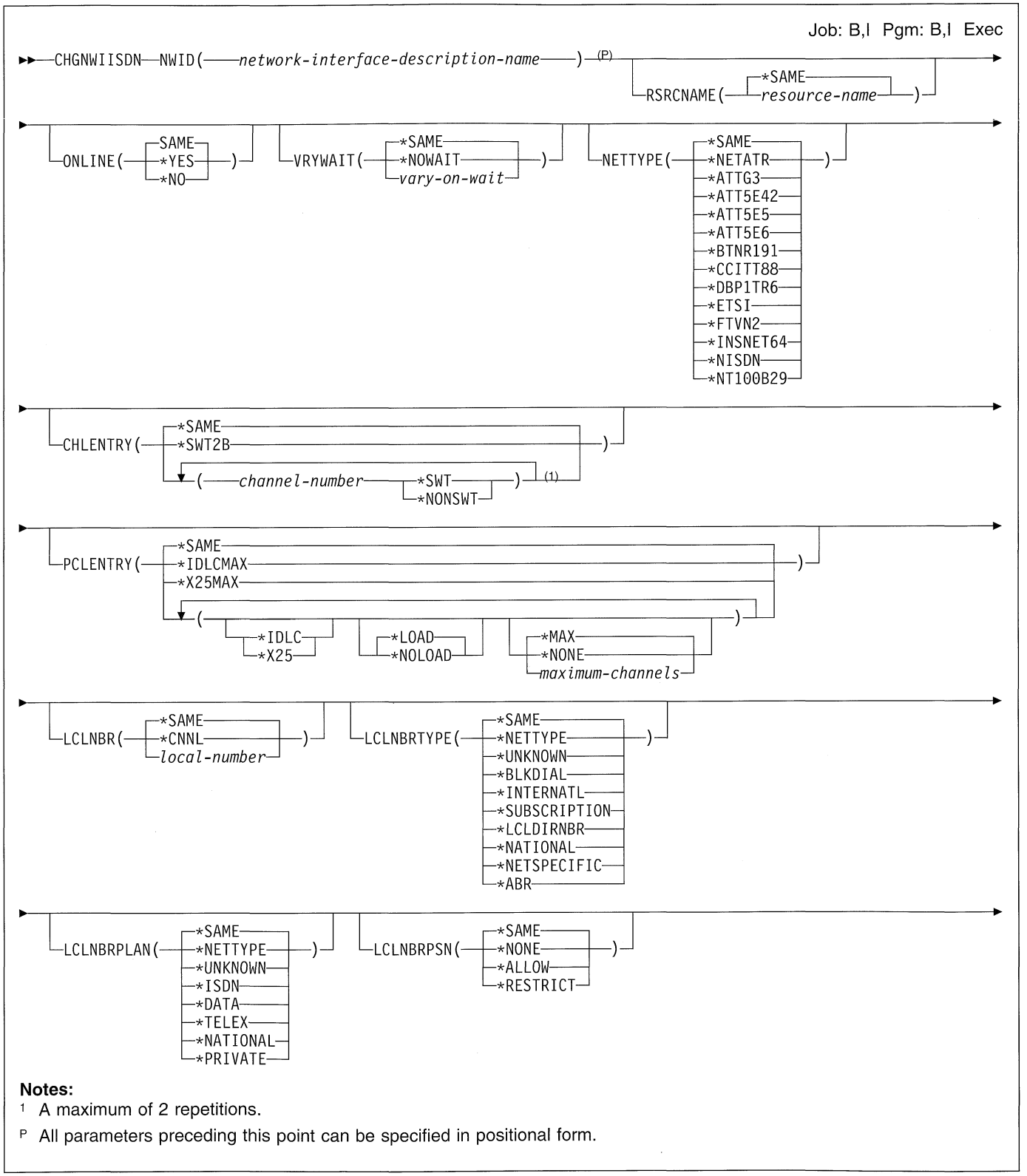
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGNWIFR   NWID(THISONE)  ONLINE(*YES)
  VRYWAIT(*NOWAIT)  CLOCK(*MODEM)
  POLLINTV(*SAME)
```

This command changes the network interface THISONE so it varies on automatically at initial program load (IPL) with no wait time. The clocking function for the network interface is changed so the modem provides the clocking. The polling interval remains the same as it was prior to this CHGNWIFR command.

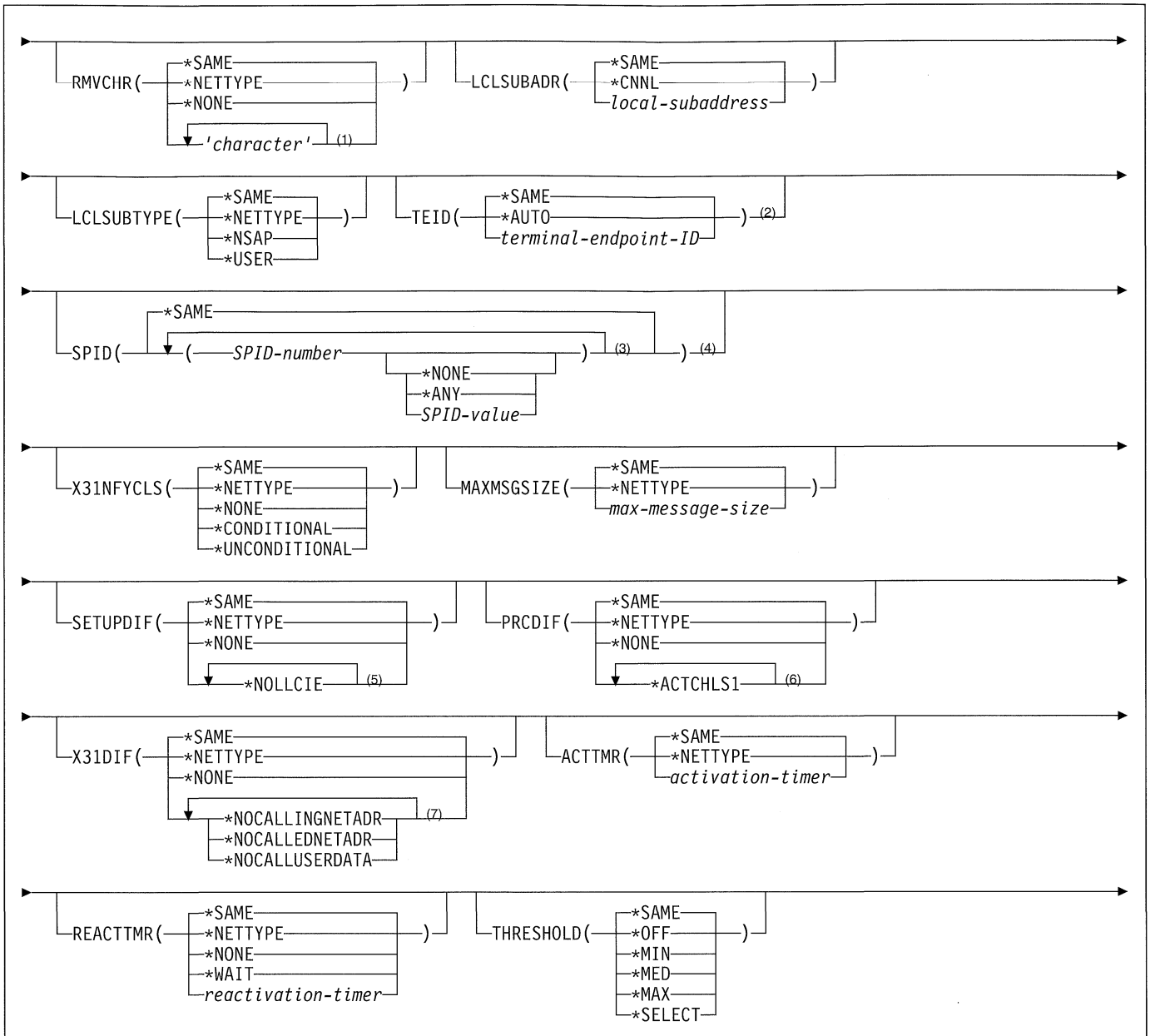
CHGNWIISDN (Change Network Interface Description for ISDN) Command



Notes:

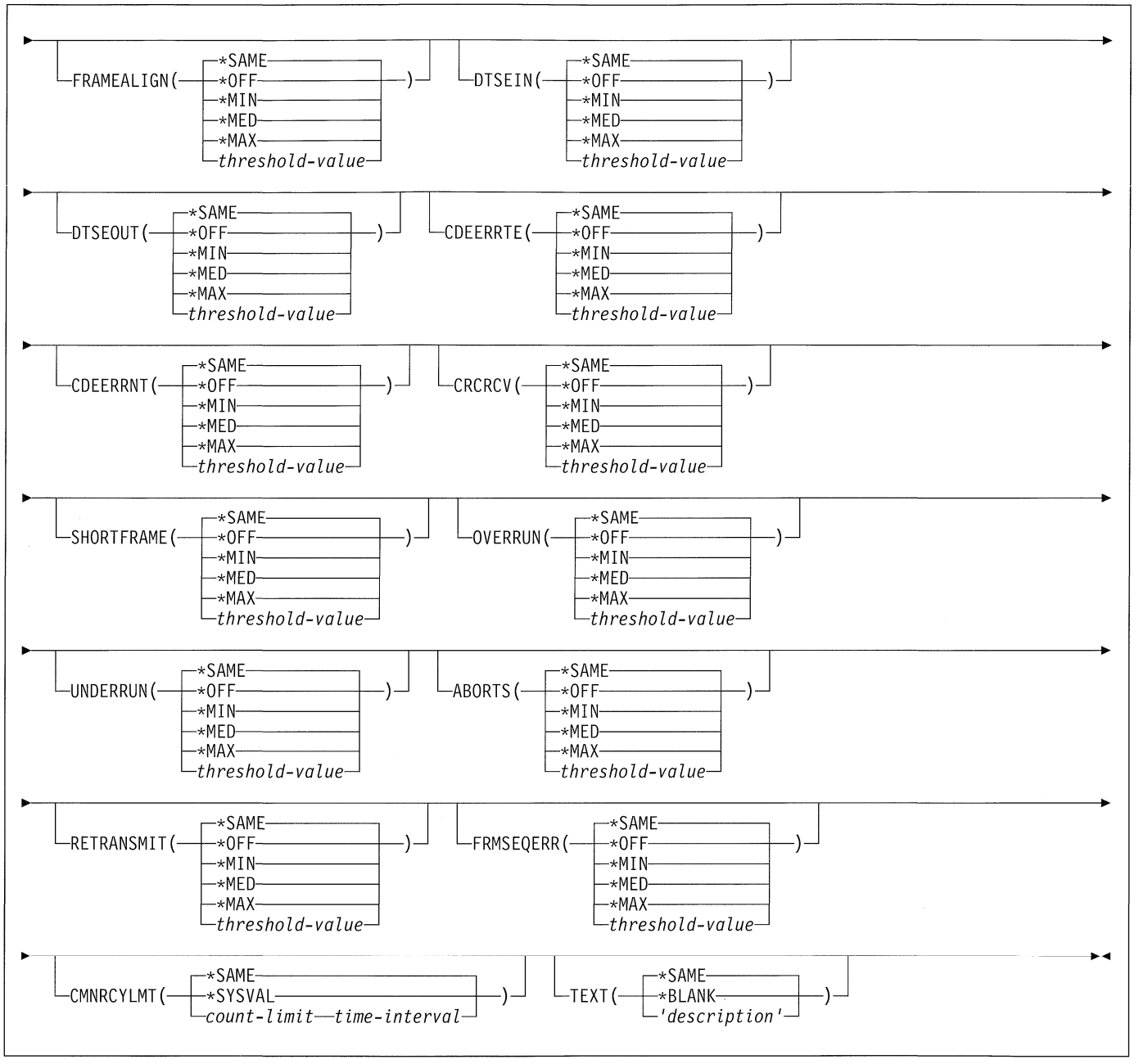
1 A maximum of 2 repetitions.

P All parameters preceding this point can be specified in positional form.



Notes:

1. A maximum of 10 repetitions.
2. This parameter is not valid when NETTYPE(*NISDN) is specified.
3. A maximum of 2 repetitions.
4. This parameter is valid only when NETTYPE(*NISDN) is specified.
5. A maximum of 4 repetitions.
6. A maximum of 6 repetitions.
7. A maximum of 3 repetitions.



Purpose

The Change Network Interface Description for ISDN (CHGNWIISDN) command changes a network interface description for an integrated services digital network (ISDN) attachment. More information about ISDN is in the *ISDN Guide*.

Required Parameter

NWID

Specifies the name of the network interface description.

Optional Parameters

RSRCNAME

Specifies the resource name that identifies the hardware that the description represents.

***SAME:** The value does not change.

resource-name: Specify the resource name.

ONLINE

Specifies whether this object is automatically varied on at initial program load (IPL).

***SAME:** The value does not change.

***YES:** The network interface is automatically varied on at IPL.

***NO:** This network interface is not automatically varied on at IPL.

VRYWAIT

Specifies whether the network interface is varied on asynchronously or synchronously. For synchronous vary on, specifies how long the system waits for the vary on to complete.

***SAME:** The value does not change.

***NOWAIT:** The system does not wait for the vary on to complete. The network interface is varied on asynchronously.

vary-on-wait: Specify the time (in seconds) to wait. Valid values range from 15 through 180. The system waits until the network interface is varied on, or until the specified time passes, before completing the Vary Configuration (VRYCFG) command.

Notes:

1. Specifying a wait time in the network interface description affects system IPL time, if ONLINE(*YES) is used, by the amount of time it takes to synchronously vary on the network interface or reach the wait-time value.
2. The time required to vary on a network interface is the time it takes to put tasks in place to manage the network interface, to activate the communications I/O processor (IOP) (including downloading the IOP model-unique Licensed Internal Code), and to establish communications with the data circuit-terminating equipment (DCE). Normal vary-on time ranges from 5 through 45 seconds, but can be longer, depending on the system, network interface protocol, and other factors.

NETTYPE

Specifies the type of integrated services digital network (ISDN) to which the system is attached. The value specified on this parameter is used to determine the defaults for several parameters that are dependent on the type of ISDN to which the system is attached.

***SAME:** The value does not change.

***NETATR:** The default network type (DFTNETTYPE) specified in the network attributes is used. The values of the parameters in the network attributes for the system can be displayed by using the Display Network Attributes (DSPNETA) command. If no value is specified on the Default Network Type parameter, the user should specify one by using the Change Network Attributes (CHGNETA) command.

***ATTG3:** This value is used when attaching to an ISDN in the United States or Canada that uses AT&T DEFINITY** G3i switching equipment.

***ATT5E42:** This value is used when attaching to an ISDN in the United States or Canada that uses AT&T 5ESS version 5E4.2 switching equipment.

***ATT5E5:** This value is used when attaching to an ISDN in the United States or Canada that uses AT&T 5ESS version 5E5 switching equipment.

***ATT5E6:** This value is used when attaching to an ISDN in the United States or Canada that uses AT&T 5ESS version 5E6 switching equipment.

***BTNR191:** This value is used when attaching to an ISDN in the United Kingdom controlled by British Telecomm.

***CCITT88:** The ISDN default values recommended by the International Telegraph and Telephone Consultative Committee (CCITT) in 1988 are used.

***DBP1TR6:** This value is used when attaching to an ISDN controlled by Germany's post telephone and telegraph administration (PTT). (Deutsche Bundespost 1TR6.)

***ETSI:** This value is used when attaching to an ISDN that uses the European Telecommunications Standards Institute (ETSI, also known as EuroISDN) standard.

***FTVN2:** This value is used when attaching to version 2 of the ISDN controlled by France's PTT. (France Telecom Numeris VN2.)

***INSNET64:** This value is used when attaching to the INSNET64 ISDN controlled by Japan's Nippon Telephone and Telegraph Public Corporation (NTT).

***NISDN:** This value is used when attaching to an ISDN that uses the National ISDN-1 or National ISDN-2 standard for North America.

***NT100B29:** This value is used when attaching to an ISDN in the United States or Canada that uses Northern Telecom DMS100 Version BCS-29 switching equipment.

CHLENTRY

Specifies a channel entry corresponding to a B channel associated with the network interface. The user can specify up to two B channel entries for a basic rate interface.

Element 1: Channel Number

***SAME:** The value does not change.

***SWT2B:** This value is used for two switched B channels.

channel-number: Specify the B-channel number. Valid values are 1 and 2.

Element 2: Channel Connection

***SWT:** This value is used for switched or semi-permanent connections.

***NONSWT:** This value is used for nonswitched permanent connections.

PCLENTRY

Specifies a list of protocols used and protocol-specific information.

***SAME:** The value does not change.

***IDLCMAX:** Specifies the IDLC protocol. The associated microcode is preloaded, and the maximum number of channels (2) are available for use.

***X25MAX:** Specifies the X.25 protocol. The associated microcode is preloaded, and the maximum number of channels (2) are available for use.

Element 1: Protocol Used

***IDLC:** The IDLC protocol is used.

***X25:** The X.25 protocol is used.

Element 2: Preload Microcode

***LOAD:** Microcode is preloaded. Preloading allows the system to report faster to incoming calls at vary-on time, since the necessary tasks are already in place.

***NOLOAD:** Microcode is not preloaded.

Element 3: Maximum Channels

***MAX:** The maximum number of channels (2) for this network interface description (NWID) are available for use by the specified protocol.

***NONE:** The maximum number of channels is not specified.

maximum-channels: Specify the maximum number of channels to be used. Valid values are 1 and 2.

LCLNBR

Specifies the local number for this system in the ISDN network. The local number can be up to 40 characters long.

***SAME:** The value does not change.

***CNNL:** The system determines the local number by using the connection list object specified for the call. Special characters are used to delimit the number; see the RMVCHR parameter.

'local-number': Specify the local number.

LCLNBRTYPE

Specifies the type of local number specified on the LCLNBR parameter.

***SAME:** The value does not change.

***NETTYPE:** The system determines the local number type by using the value specified on the NETTYPE parameter.

***UNKNOWN:** The local number type is not known.

***BLKDIAL:** The local number is a block dial type.

***INTERNATL:** The local number is an international number type.

***SUBSCRIPTION:** The local number is a subscription number type.

***LCLDIRNBR:** The local number is a local directory number type.

***NATIONAL:** The local number is a national address type.

***NETSPECIFIC:** The local number type is specific to the network.

***ABR:** The local number type is abbreviated.

LCLNBRPLAN

Specifies the numbering plan used for the local number.

***SAME:** The value does not change.

***NETTYPE:** The numbering plan is determined by the value specified on the NETTYPE parameter.

***UNKNOWN:** The numbering plan is not known.

***ISDN:** The ISDN/telephony numbering plan is used.

***DATA:** The data numbering plan is used.

***TELEX:** The telex numbering plan is used.

***NATIONAL:** The national numbering plan is used.

***PRIVATE:** A private numbering plan is used.

LCLNBRPSN

Specifies whether the calling user presents the local number to the called user. This parameter applies only to outgoing calls.

***SAME:** The value does not change.

***NONE:** The local number presentation is not encoded. The network determines whether the local number is presented to the called user.

***ALLOW:** The local number is presented to the called user.

***RESTRICT:** The presentation of the local number to the called user is restricted by the network.

RMVCHR

Specifies up to 10 characters to be removed from local number before the number is used by the system. Extra characters are removed from numbers before sending or comparing the numbers. The ability to shorten numbers prior to their use by the system means the user can insert extra characters in numbers to help make them more organized and readable.

***SAME:** The value does not change.

***NETTYPE:** The system determines the characters to be removed using the value specified on the NETTYPE parameter.

***NONE:** No characters are removed.

'character': Specify up to 10 characters to be removed.

LCLSUBADR

Specifies the local subaddress assigned to a system configured as a terminal endpoint. The subaddress can be up to 40 hexadecimal characters in length.

***SAME:** The value does not change.

***CNNL:** The system determines the local number field by using the connection list object specified for the call.

local-subaddress: Specify the local subaddress.

LCLSUBTYPE

Specifies the type of subaddress assigned to a terminal endpoint system.

***SAME:** The value does not change.

***NETTYPE:** The subtype is determined by the value specified on the NETTYPE parameter.

***NSAP:** The local subaddress is NSAP (network-layer service access point).

***USER:** The local subaddress is user-specified.

TEID

Specifies the terminal endpoint identifier (TEID).

Note: The TEID parameter is not available and the TEID is automatically assigned for network interface descriptions with *NISDN specified on the NETTYPE parameter.

***SAME:** The value does not change.

***AUTO:** A terminal endpoint identifier is automatically assigned by the network.

terminal-endpoint-identifier: Specify a number to use as the terminal endpoint identifier. Valid values range from 0 through 63.

SPID

Specifies the service profile identifier (SPID) to be used in an exchange of information between the system and the network when communications are initialized. The SPID value is assigned by and can be obtained from the network provider when the user subscribes to the network.

***SAME:** The value does not change.

Element 1: SPID Number

SPID-number: Specify the SPID number. Valid values are 1 and 2.

Element 2: SPID Value

***NONE:** No SPID or TEID is used for the exchange.

***ANY:** An SPID is not specified, but a TEID is used for the exchange.

SPID-value: Specify the service profile identifier. A minimum of 9 characters are required, and no more than 20 characters can be specified.

X31NFYCLS

Specifies the method for the packet handler in the ISDN to notify the system of an incoming packet mode call.

***SAME:** The value does not change.

***NETTYPE:** The method of notifying the system is determined by the value specified on the NETTYPE parameter.

***NONE:** The packet handler provides no notification.

***CONDITIONAL:** The packet handler provides notification only if a packet mode call cannot be delivered on an existing connection.

***UNCONDITIONAL:** The packet handler provides notification for every packet mode call.

MAXMSGSIZE

Specifies the maximum size of a Q.931 message to be accepted by the call control layer in the I/O processor.

***SAME:** The value does not change.

***NETTYPE:** The system determines the maximum size by using the value specified on the NETTYPE parameter.

max-message-size: Specify the maximum message size. Valid values range from 110 through 1024.

Note: Use of the default value is strongly recommended for this parameter; values less than 262 may result in data loss.

SETUPDIF

Specifies the defaults used to send **SETUP** messages to the network. Up to four values can be specified on this parameter. The user can follow International Telegraph and Telephone Consultative Committee (CCITT) recommendations by specifying the *NONE value.

***SAME:** The value does not change.

***NETTYPE:** The system determines the defaults for sending **SETUP** messages by using the value specified on the NETTYPE parameter.

***NONE:** The CCITT recommended defaults for sending **SETUP** messages are used: High-Layer Compatibility IE is not sent; Low-layer Compatibility IE is sent; Bearer Capability IE is sent in the **SETUP** message; and Terminal Capability IE is not sent in the **SETUP** message.

***NOLLCIE:** The Low-Layer Compatibility IE is not sent with the **SETUP** message. **SETUP** message.

PRCDIF

Specifies the defaults used for the handling of miscellaneous procedures. More than one value can be specified on this parameter. The user can follow International Telegraph and Telephone Consultative Committee (CCITT) recommendations by specifying the *NONE value.

***SAME:** The value does not change.

***NETTYPE:** The system determines miscellaneous settings by using the value specified on the NETTYPE parameter.

***NONE:** The CCITT recommended defaults for miscellaneous settings are used: Calls are cleared when a **SETUP ACKNOWLEDGE** is received; a **STATUS** message is sent to the peer when an error is detected; both sides send **RELEASE** messages following a **DISCONNECT** message; **RELEASE COMPLETE** messages are sent to callers.

***ACTCHLS1:** The S1 maintenance channel is activated. The S1 channel allows the system to accept a broader range of RAS messages, improving the system's network management capabilities.

X31DIF

Specifies network-specific differences in X.31 Case B operations.

***SAME:** The value does not change.

***NETTYPE:** The system determines whether X.31 call data is checked depending on the value specified on the NETTYPE parameter.

***NONE:** There are no differences in packet mapping.

***NOCALLINGNETADR:** The call data does not contain a calling network address.

***NOCALLEDNETADR:** The call data does not contain a called network address.

***NOCALLUSERDATA:** The call data does not contain user data.

ACTTMR

Specifies the amount of time allowed in tenths of a second, for the interface activation process to complete.

***SAME:** The value does not change.

***NETTYPE:** The system determines the amount of time allowed by using the value specified on the NETTYPE parameter.

activation-timer: Specify a value ranging from 10 through 300 in 0.1-second intervals.

REACTTMR

Specifies the amount of time allowed in tenths of a second, for interface reactivation to occur following a temporary loss of synchronization.

***SAME:** The value does not change.

***NETTYPE:** The system determines the amount of time allowed by using the value specified on the NETTYPE parameter.

***NONE:** The system does not wait for reactivation to occur.

***WAIT:** The system waits indefinitely for reactivation to occur.

reactivation-timer: Specify a value ranging from 10 through 1800 in 00.1-second intervals.

THRESHOLD

Specifies the temporary error threshold level being monitored by the system. A permanent error is reported only if the errors occurred consecutively and exceeded the retry limit.

Note: Specifying the THRESHOLD parameter affects all threshold errors. They cannot be specified individually.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** The error threshold is set at a minimum monitoring level.

***MED:** The system performs a medium amount of error threshold monitoring for all types of errors. The values chosen are different for each threshold error type. See the descriptions on the next eleven THRESHOLD parameters for more specific information.

***MAX:** The error threshold is set at a maximum monitoring level.

***SELECT:** Allows each of the following error threshold parameters to be specified individually. The next eleven parameters are the threshold parameters. They are: Loss of frame alignment, Incoming system access errors, Outgoing system access errors, Code error detected by TE, Code error detected by NT, CRC errors received, Overrun, Underrun, Frame aborts, Retransmitted frames, and Frame sequence errors.

FRAMEALIGN

Specifies the threshold for the number of frame alignment errors beyond which a message is sent informing the user that errors have occurred.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** The system performs the minimum amount of error threshold monitoring for frame alignment errors: 9 errors in the first 30 seconds or 270 errors in any 15 minutes. The system is more tolerant of errors than if *MED or *MAX is specified.

***MED:** The system performs a medium amount of error threshold monitoring for frame alignment errors: 3 errors in the first 30 seconds or 90 errors in any 15 minutes.

***MAX:** The system performs the maximum amount of error threshold monitoring for frame alignment errors.

threshold-value: Specify a number, ranging from 1 through 10000, that corresponds to the number of errors allowed in a 15-minute interval.

DTSEIN

Specifies the threshold for the number of incoming system access errors beyond which a message is sent informing the user that errors have occurred.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** The system performs the minimum amount of error threshold monitoring for incoming system access errors: 9 errors in the first 30 seconds or 270 errors in any 15 minutes. The system is more tolerant of errors than if *MED or *MAX is specified.

***MED:** The system performs a medium amount of error threshold monitoring for incoming system access errors: 3 errors in the first 30 seconds or 90 errors in any 15 minutes.

***MAX:** The system performs the maximum amount of error threshold monitoring for incoming system access errors.

threshold-value: Specify a number, ranging from 1 through 5000, that corresponds to the number of errors allowed in a 15-minute interval.

DTSEOUT

Specifies the threshold for the number of outgoing system access errors beyond which a message is sent informing the user that errors have occurred.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** The system performs the minimum amount of error threshold monitoring for outgoing system access errors: 9 errors in the first 30 seconds or 270 errors in any 15 minutes. The system is more tolerant of errors than if *MED or *MAX is specified.

***MED:** The system performs a medium amount of error threshold monitoring for outgoing system access errors: 3 errors in the first 30 seconds or 90 errors in any 15 minutes.

***MAX:** The system performs the maximum amount of error threshold monitoring for outgoing system access errors.

threshold-value: Specify a number, ranging from 1 through 5000, that corresponds to the number of errors allowed in a 15-minute interval.

CDEERRTE

Specifies the threshold for the number of code errors detected by the terminal endpoint (TE) beyond which a message is sent informing the user that errors have occurred.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** The system performs the minimum amount of error threshold monitoring for code errors detected by the TE: 9 errors in the first 30 seconds or 270 errors in any 15 minutes. The system is more tolerant of errors than if *MED or *MAX is specified.

***MED:** The system performs a medium amount of error threshold monitoring for code errors detected by the TE: 3 errors in the first 30 seconds or 90 errors in any 15 minutes.

***MAX:** The system performs the maximum amount of error threshold monitoring for code errors detected by the TE.

threshold-value: Specify a number, ranging from 1 through 10000, that corresponds to the number of errors allowed in a 15-minute interval.

CDEERRNT

Specifies the threshold for the number of code errors detected by the network termination (NT) beyond which a message is sent informing the user that errors have occurred.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** The system performs the minimum amount of error threshold monitoring for code errors detected by the NT: 9 errors in the first 30 seconds or 270 errors in any 15 minutes. The system is more tolerant of errors than if *MED or *MAX is specified.

***MED:** The system performs a medium amount of error threshold monitoring for code errors detected by the NT: 3 errors in the first 30 seconds or 90 errors in any 15 minutes.

***MAX:** The system performs the maximum amount of error threshold monitoring for code errors detected by the NT.

threshold-value: Specify a number, ranging from 1 through 5000, that corresponds to the number of errors allowed in a 15-minute interval.

CRCRCV

Specifies the threshold for the number of Cyclic Redundancy Check (CRC) errors received beyond which a message is sent informing the user that errors have occurred.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** The system performs the minimum amount of error threshold monitoring for CRC errors received: 5 errors in the first 30 seconds or 150 errors in any 15 minutes. The system is more tolerant of errors than if *MED or *MAX is specified.

***MED:** The system performs a medium amount of error threshold monitoring for CRC errors received: 2 errors in the first 60 seconds or 30 errors in any 15 minutes.

***MAX:** The system performs the maximum amount of error threshold monitoring for CRC errors received.

threshold-value: Specify a number, ranging from 1 through 10000, that corresponds to the number of errors allowed in a 15-minute interval.

SHORTFRAME

Specifies the threshold for the level of errors for short frames received beyond which a message is sent informing the user that errors have occurred.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** The system performs the minimum amount of error threshold monitoring for receive overrun errors: 6 short X.25 or SDLC frames received in the first 30 seconds or 1 short frame received every second for 14 minutes. The system is more tolerant of errors than if *MED or *MAX is specified.

***MED:** The system performs a medium amount of error threshold monitoring for receive overrun errors: 6 X.25 or SDLC frames received in the first 30 seconds or 1 received every 3-4 seconds for 10-14 minutes.

***MAX:** The system performs the maximum amount of error threshold monitoring for short frame errors.

threshold-value: Specify a number, ranging from 1 through 10000, that corresponds to the number of errors allowed in a 15-minute interval.

OVERRUN

Specifies the threshold for the number of receive overrun errors beyond which a message is sent informing the user that errors have occurred.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** The system performs the minimum amount of error threshold monitoring for receive overrun errors: 2 errors in the first 90 seconds or 20 errors in any 15 minutes. The system is more tolerant of errors than if *MED or *MAX is specified.

***MED:** The system performs a medium amount of error threshold monitoring for receive overrun errors: 2 errors in the first 300 seconds or 6 errors in any 15 minutes.

***MAX:** The system performs the maximum amount of error threshold monitoring for receive overrun errors.

threshold-value: Specify a number, ranging from 1 through 3000, that corresponds to the number of errors allowed in a 15-minute interval.

UNDERRUN

Specifies the threshold for the number of transmit underrun errors beyond which a message is sent informing the user that errors have occurred.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** The system performs the minimum amount of error threshold monitoring for buffer overrun errors: 2 errors in the first 90 seconds or 20 errors in any 15 minutes. The system is more tolerant of errors than if *MED or *MAX is specified.

***MED:** The system performs a medium amount of error threshold monitoring for buffer overrun errors: 2 errors in the first 300 seconds or 6 errors in any 15 minutes.

***MAX:** The system performs the maximum amount of error threshold monitoring for transmit underrun errors.

threshold-value: Specify a number, ranging from 1 through 3000, that corresponds to the number of errors allowed in a 15-minute interval.

ABORTS

Specifies the threshold for the number of frame aborts beyond which a message is sent informing the user that errors have occurred.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** The system performs the minimum amount of error threshold monitoring for frame aborts: 5 errors in the first 30 seconds or 150 errors in any 15-minute interval. The system is more tolerant of errors than if *MED or *MAX is specified.

***MED:** The system performs a medium amount of error threshold monitoring for frame aborts: 2 errors in the first 60 seconds or 30 errors in any 15 minutes.

***MAX:** The system performs the maximum amount of error threshold monitoring for frame aborts.

threshold-value: Specify a number, ranging from 1 through 5000, that corresponds to the number of errors allowed in a 15-minute interval.

RETRANSMIT

Specifies the threshold for the number of retransmissions of frames beyond which a message is sent informing the user that errors have occurred.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** The system performs the minimum amount of error threshold monitoring for retransmitted frames aborted: 5 errors in the first 30 seconds or 150 errors in any 15 minutes. The system is more tolerant of errors than if *MED or *MAX is specified.

***MED:** The system performs a medium amount of error threshold monitoring for retransmitted frames aborted: 2 errors in the first 60 seconds or 30 errors in any 15 minutes.

***MAX:** The system performs the maximum amount of error threshold monitoring for retransmitted frames aborted.

threshold-value: Specify a number, ranging from 1 through 10000, that corresponds to the number of errors allowed in a 15-minute interval.

FRMSEQERR

Specifies the threshold for the number of frame sequence errors beyond which a message is sent informing the user that errors have occurred.

***SAME:** The value does not change.

***OFF:** No monitoring of errors occurs.

***MIN:** The system performs the minimum amount of error threshold monitoring for frame sequence errors: 2 errors in the first 90 seconds or 20 errors in any 15 minutes. The system is more tolerant of errors than if *MED or *MAX is specified.

***MED:** The system performs a medium amount of error threshold monitoring for frame sequence errors: 2 errors in the first 300 seconds or 6 errors in any 15 minutes.

***MAX:** The system performs the maximum amount of error threshold monitoring for frame sequence errors.

threshold-value: Specify a number, ranging from 1 through 3000, that corresponds to the number of errors allowed in a 15-minute interval.

CMNRCYLMT

Specifies the number of recovery attempts made by the system before an inquiry message is sent to the system operator. Also specifies the time (in minutes) that must elapse before the system sends an inquiry message to the system operator indicating that the recovery attempt count limit is reached.

***SAME:** The value does not change.

***SYSVAL:** The recovery limits specified in the QCMNRCYLMT system value are used.

Element 1: Count Limit

count-limit: Specify the number of recovery attempts to be made. Valid values range from 0 through 99.

Element 2: Time Interval

time-interval: Specify the time interval (in minutes) at which the specified number of second-level recoveries are attempted. Valid values range from 0 through 120.

If the value specified for *count-limit* is not 0, the value 0 specifies infinite recovery.

TEXT

Specifies text that briefly describes the network interface description. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

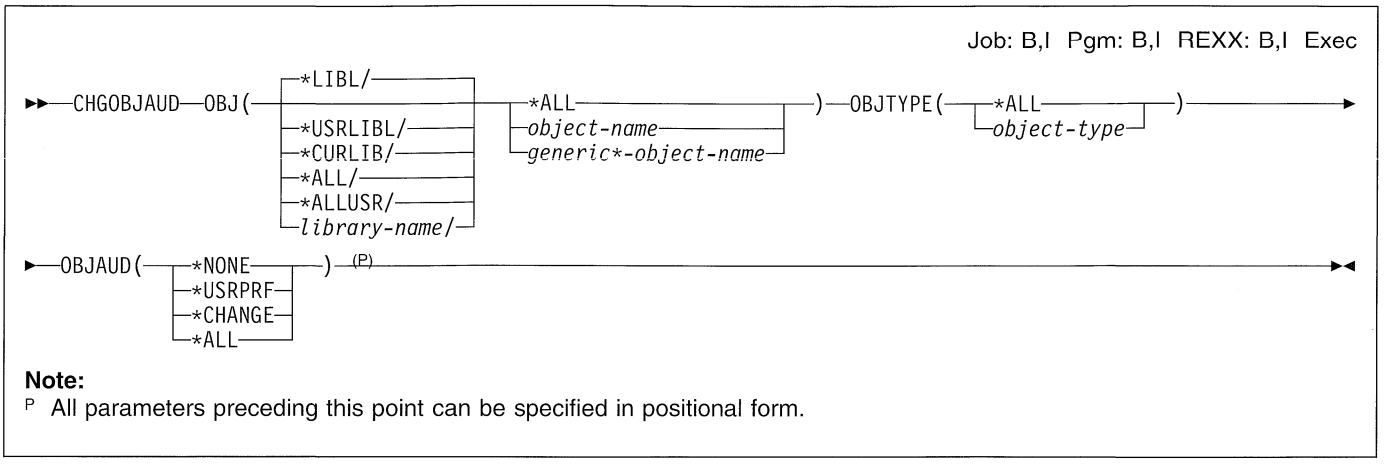
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGNWIISDN   NWID(ISDNNET)  RSRCNAME(LIN021)
```

This command changes the resource name for the network interface described name ISDNNET to LIN021.

CHGOBJAUD (Change Object Auditing) Command



Note:
P All parameters preceding this point can be specified in positional form.

Purpose

The Change Object Auditing (CHGOBJAUD) command allows users with *AUDIT special authority to set up or change auditing on an object. Users with *AUDIT special authority can turn auditing on or off for an object regardless of whether they have authority to the object. The system value QAUDCTL controls turning auditing on and off. The auditing attribute of an object can be displayed with the Display Object Description (DSPOBJD) command.

Required Parameters

OBJ

Specifies the name of the object for which auditing values are being changed.

The name of the object can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

***USRLIBL:** Only the libraries in the user portion of the job's library list are searched.

***ALL:** All libraries on the system are searched.

***ALLUSR:** All user libraries are searched. All libraries with names that do not begin with the letter Q are searched except for the following:

```
#CGULIB    #DFULIB    #RPGLIB    #SEULIB
#COBLIB    #DSULIB    #SDALIB
```

Although the following Qxxx libraries are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are considered *user libraries*, and are also searched:

```
QDSNX      QPFRDATA   QUSER38
QGPL       QRCL       QUSRSYS
QGPL38     QS36F      QUSRVxRxMx
```

Note: A different library name, of the form QUSRVxRxMx, is added with each release. VxRxMx is the version, release, and modification level of the library.

library-name: Specify the name of the library to be searched.

***ALL:** All objects that have the specified object type are changed.

object-name: Specify the name of the object for which auditing values are changed.

generic-object-name:* Specify the generic name of the object. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be changed only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

OBJTYPE

Specifies the type of the object being changed. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***ALL:** All object types that have the specified object name are changed.

object-type: Specify the type of the object for which auditing values are changed.

OBJAUD
Specifies the object auditing value for this object.

***NONE:** Using or changing this object will not cause an audit entry to be sent to the security journal.

***USRPRF:** The user profile of the user accessing this object is used to determine if an audit record will be sent for this access. The OBJAUD keyword of the CHGUSRAUD command is used to turn on auditing for a specific user.

***CHANGE:** All change accesses to this object by all users are logged.

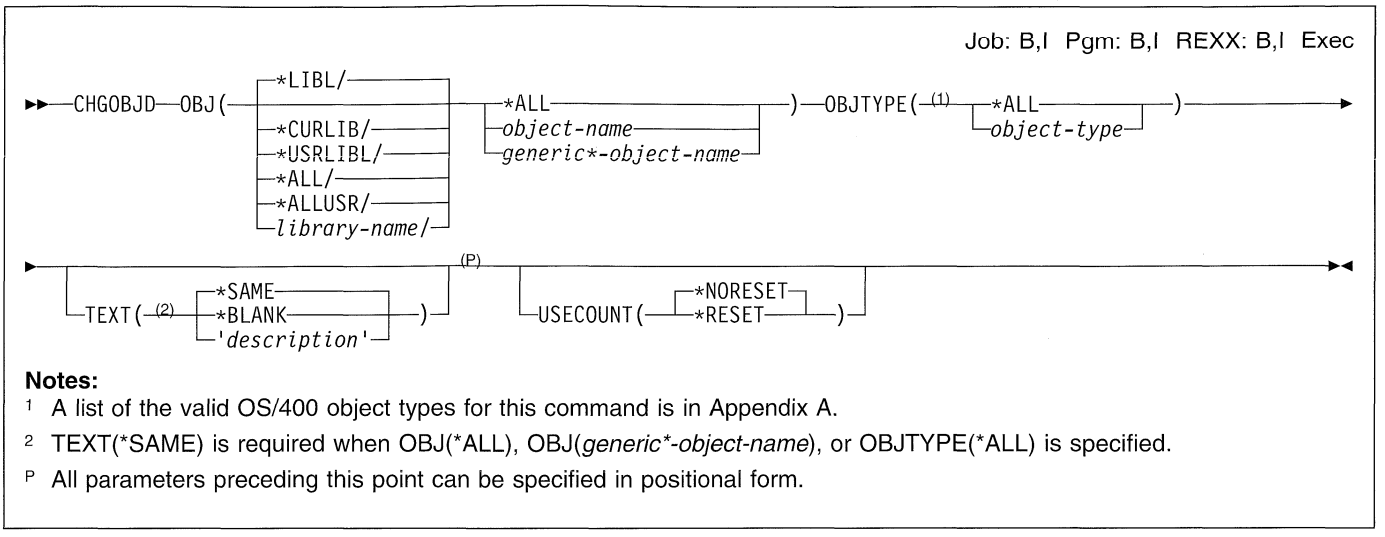
***ALL:** All change or read accesses to this object by all users are logged.

Example

```
CHGOBJAUD  OBJ(PAYROLL/PAYFILE)  OBJTYPE(*FILE)
  OBJAUD(*CHANGE)
```

This command changes the object auditing value of the PAYFILE object in the PAYROLL library that has an object type *FILE. The auditing value of the PAYFILE file is changed so that changes to the file by any user is logged to the auditing journal QAUDJRN in QSYS.

CHGOBJD (Change Object Description) Command



Notes:

1 A list of the valid OS/400 object types for this command is in Appendix A.

2 TEXT(*SAME) is required when OBJ(*ALL), OBJ(*generic*-object-name*), or OBJTYPE(*ALL) is specified.

P All parameters preceding this point can be specified in positional form.

Purpose

The Change Object Description (CHGOBJD) command changes the text description of an object and resets the days used count for the object to zero.

Restriction: Users of this command must have object operational and object management authority to the object being changed and *READ authority to the library where the object is located.

Required Parameters

OBJ

Specifies the qualified name of the object whose text description is being changed. If no library name is given, *LIBL is used to find the specified object. The library name is entered to ensure that the correct object is being specified.

Note: Users changing the object description for objects of type *DEVD should first use the Allocate Object (ALCOBJ) command to obtain an *EXCLRD lock state on the device description. If this is not done, a CPF2114 message is sent, which indicates that the device description is locked. The ALCOBJ command does not have to be used if the user is already signed on the display device for which the description is being changed.

The name of the object can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

***USRLIBL:** Only the libraries in the user portion of the job's library list are searched.

***ALL:** All libraries in the system portion of the job's library list, including QSYS, are searched.

***ALLUSR:** All user libraries are searched. All libraries with names that do not begin with the letter Q are searched except for the following:

```
#CGULIB    #DFULIB    #RPGLIB    #SEULIB
#COBLIB    #DSULIB    #SDALIB
```

Although the following Qxxx libraries are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are considered *user libraries*, and are also searched:

```
QDSNX      QPFRDATA   QUSER38
QGPL       QRCL       QUSRSYS
QGPL38     QS36F      QUSRVxRxMx
```

Note: A different library name, of the form QUSRVxRxMx, is added with each release. VxRxMx is the version, release, and modification level of the library.

library-name: Specify the name of the library to be searched.

***ALL:** All objects in the specified library (the library qualifier for this parameter) of the type specified on the OBJTYPE parameter are changed.

object-name: Specify the name of the object that is changed.

generic-object-name:* Specify the generic name of the object. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete

object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be changed only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

OBJTYPE
Specifies the object type of the object that is being changed. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***ALL:** All objects in the specified library (the library qualifier for the OBJ parameter) are to be changed.

object-type: Specify the name of the object type to be changed.

Optional Parameters

TEXT
Specifies text that briefly describes the object and its function. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

USECOUNT
Specifies whether the days-used count for the object is reset.

***NORESET:** The days-used count for the object is not reset.

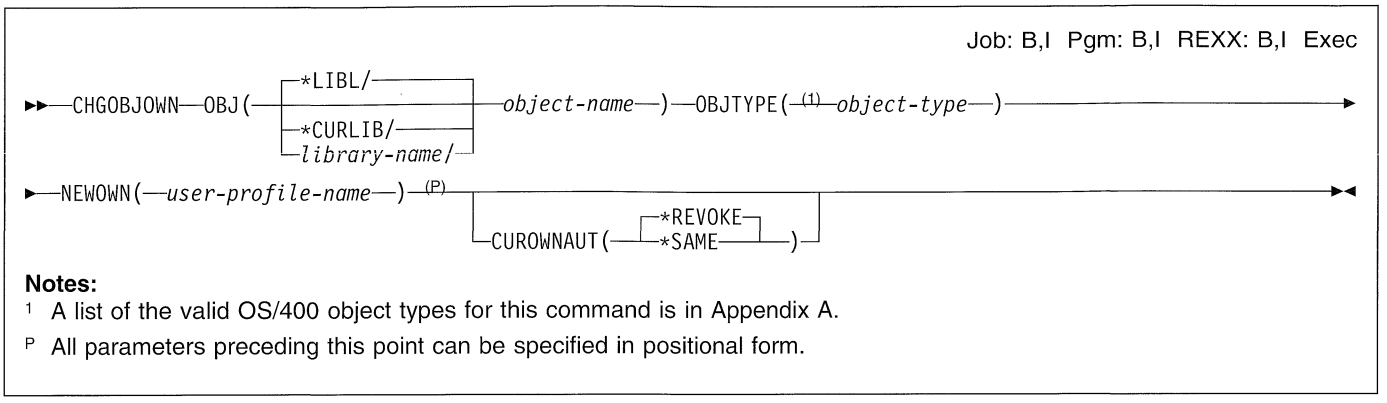
***RESET:** The days used count for the object is reset to zero.

Example

```
CHGOBJD  OBJ(LIB1/DA1)  OBJTYPE(*DTAARA)
  TEXT('NEW TEXT FOR DA1')
```

This command changes the text description of the data area named DA1 located in the library LIB1 to the value specified by the TEXT parameter.

CHGOBJOWN (Change Object Owner) Command



Notes:
1 A list of the valid OS/400 object types for this command is in Appendix A.
P All parameters preceding this point can be specified in positional form.

Purpose

The Change Object Owner (CHGOBJOWN) command transfers object ownership from one user to another. The authority that other users have to the object does not change.

The owner of an object always has all the authorities applicable to the object unless they are explicitly revoked. The owner of an object has the authority to grant any authorities to any user for that object. Owners can also grant to themselves authorities that were previously revoked. Owners may, for example, remove some of their specific authorities as a precautionary measure, and then, when the need arises, grant those same authorities to themselves again.

The security officer has complete authority for all objects and can transfer the ownership of any object. All users have add and delete authorities for their own user profiles; that is, users can add objects to or delete objects (that they created) from their own user profiles by transferring the ownership of the object.

Restrictions:

1. To transfer ownership, you must have all of the following:
   - Object existence authority for the object
   - Object operational and object existence authorities if the object is a file, library, or subsystem description
   - All object authority or ownership if the object is an authorization list
   - Add authority for the new owner's user profile
   - Delete authority for the present owner's user profile
2. For display stations, if this command is not entered at the device whose ownership is being changed or whose message queue's ownership is being changed, this command should be preceded by the Allocate Object (ALCOBJ) command and followed by the Deallocate Object (DLCOBJ) command.
3. Object type *DOC or *FLR cannot be specified; the user must use DLO (document interchange) support.
4. Changing the ownership of an object that has an authority holder associated with it also changes the ownership of the authority holder.
5. You must have all object and security administrator authorities to change the object owner of a program or an SQL (Structured Query Language) package that adopts authority.

Required Parameters

OBJ
Specifies the qualified name of the object that is being assigned to the new owner. The library name can be entered to ensure that the correct object changes ownership.

The name of the object can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

object-name: Specify the name of the object that is assigned.

OBJTYPE
Specifies the object type, such as program (*PGM), file (*FILE), or library (*LIB) of the object whose ownership is being transferred. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

NEWOWN
Specifies the name of the user to whom the object is being assigned. The user profile must already exist.

Optional Parameter

CUROWNAUT

Specifies whether the current owner's authority is revoked when ownership is transferred to the new owner specified in the NEWOWN parameter.

***REVOKE:** The current owner's authority is revoked when the object is transferred to the new owner.

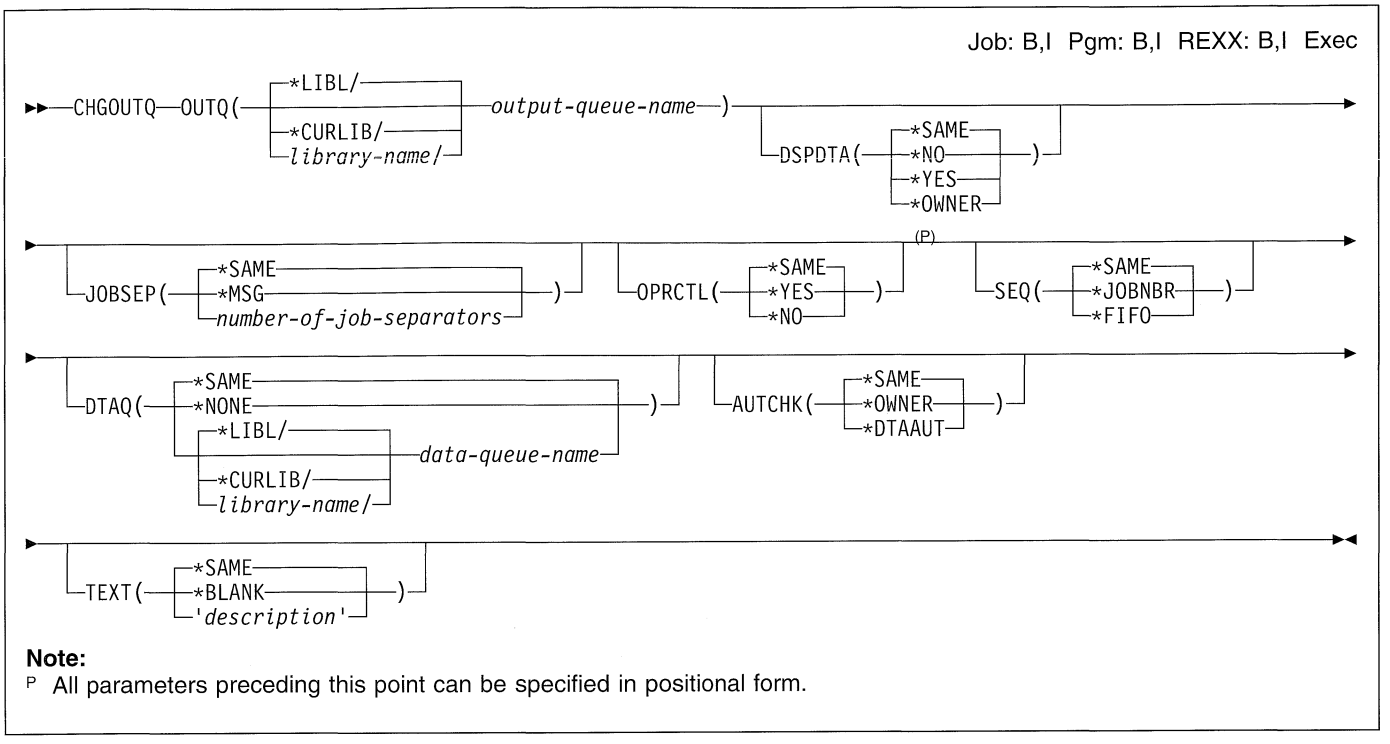
***SAME:** The value does not change.

Example

```
CHGOBJOWN  OBJ(USERLIB/PROGRAM1)  OBJTYPE(*PGM)
  NEWOWN(ANN)
```

This command assigns ownership of the program named PROGRAM1, located in the user library named USERLIB, to the user named ANN. The authority is revoked from the current owner.

CHGOUTQ (Change Output Queue) Command



Note:

| P All parameters preceding this point can be specified in positional form.

Purpose

The Change Output Queue (CHGOUTQ) command allows the user to change the attributes of the specified output queue. The attributes of the output queue, except the number of job separators, and the order of files on the output queue, can be changed while a writer is producing spooled files from the output queue. The following attributes can be changed on the specified output queue:

- Authority to display data from this output queue
- Operator's authority to control this output queue
- Number of separator pages per job
- The order of spooled files on this output queue
- Object authority required to control this output queue
- The description of this output queue
- The data queue of the output queue

If the output queue contains spooled files, the user cannot change the SEQ parameter for the queue. Other CHGOUTQ parameters such as DSPDTA, JOBSEP, and OPRCTL can normally be changed if the output queue contains spooled files. If the user attempts to change SEQ, none of the changes are allowed.

Required Parameter

OUTQ

Specifies the qualified name of the output queue whose attributes are being changed.

| The name of the output queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

output-queue-name: Specify the name of the output queue.

Optional Parameters

DSPDTA

Specifies whether users that have authority to read the output queue can display the data from any output file on the queue or display only data from their own files.

***SAME:** The value does not change.

***NO:** Users authorized to use the queue can display, copy, or send only the output data of their files (unless they have some special authority).

***YES:** Users with read authority for the output queue can display, copy, or send the output of any file on the queue.

***OWNER:** Users authorized to use the queue can display, copy, or send the output data of their own files only, unless they have *SPLCTL special authority.

JOBSEP

Specifies, for each job having spooled files on this output queue, the number of separators to be printed at the beginning of the printed output for each job. Each separator contains information that identifies the job such as its name, the job user's name, the job number, and the time and date when the job was run. The number of separators range from 0 through 9. This parameter can be changed only if the output queue is not being processed by a writer.

***SAME:** The value does not change.

***MSG:** No job separators are placed before each job's output. A message is sent to a message queue notifying the operator of the end of each job. The message queue receiving the message is identified by the MSGQ parameter on the command that started the writer.

number-of-job-separators: Specify the number of separators printed before the output of each job. Valid values range from 0 through 9.

OPRCTL

Specifies whether a user with job control authority is allowed to control and make changes to spooled files with entries on this output queue. A user has job control authority if SPCAUT(*JOBCTL) is specified in the user profile.

***SAME:** The value does not change.

***YES:** Users with job control authority can control the queue and make changes to the entries on the queue.

***NO:** This queue and its entries cannot be manipulated or changed by users with job control authority unless they have some other special authority.

SEQ

Specifies the order of the spooled files on the output queue.

***SAME:** The value does not change.

***JOBNBR:** Within priority, the queue entries for spooled files are sorted using the job number of the job that created the spooled file. (The job number is a combination of the date and time which the job entered the system.)

If *JOBNBR is specified on the SEQ parameter, spooled files of a job with the same output priority and status are grouped together.

Also if *JOBNBR is specified on the SEQ parameter, the first job always has its spooled files produced first for jobs of equal priority.

***FIFO:** The queue is first-in first-out within priority for each file. That is, new spooled files are placed after all other entries on the queue of the same priority. The following changes place a queue entry after all others of equal priority on the queue.

- A change of output priority with the Change Job (CHGJOB) command
- A change in status from held (HLD), saved (SAV), closed (CLO), or open (OPN) to available (RDY)
- A change in status from available (RDY) to not available (HLD, SAV, CLO, OPN)
- A spooled file added to the queue if the file is opened
- Using the Change Spooled File Attributes (CHGSPLFA) command to move a spooled file to an output queue which has SEQ(*FIFO) specified

DTAQ

Specifies the name of the data queue associated with the output queue.

***SAME:** The value does not change.

***NONE:** No data queue is associated with the output queue.

The name of the data queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

data-queue-name: Specify the name of the data queue associated with the output queue.

AUTCHK

Specifies whether the commands that check the requester's authority to the output queue also check for ownership authority or data authority.

***SAME:** The value does not change.

***OWNER:** The requester must have ownership authority to the output queue to pass the output queue authorization test. The requester can have ownership authority by being the owner of the output queue, sharing a group profile with the queue owner, or running a program that adopts the owner's authority.

***DTAAUT:** The requester must have the appropriate data authority to the output queue (*READ, *ADD, and *DELETE) to pass the output queue authorization test.

TEXT

Specifies text that briefly describes the output queue. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

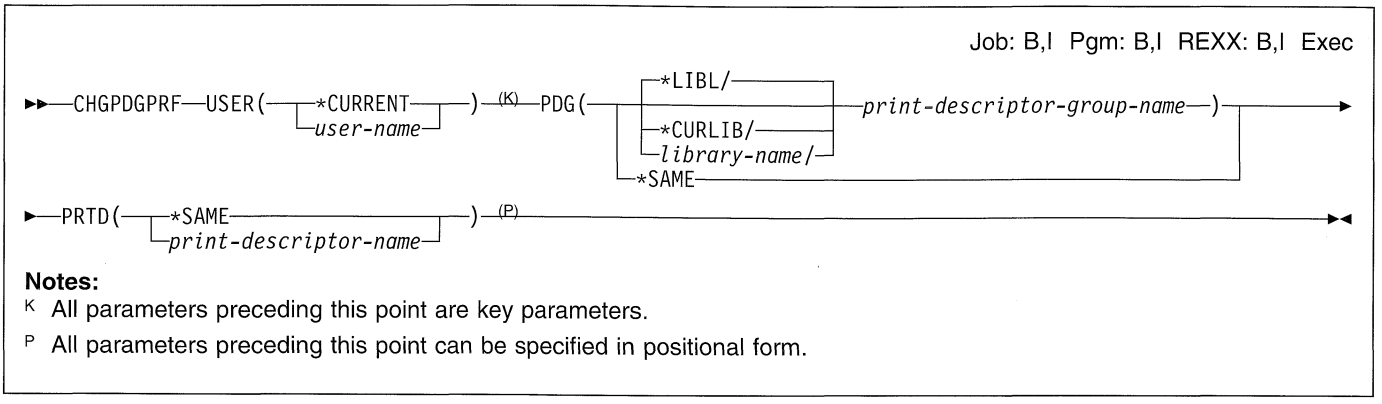
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGOUTQ   OUTQ(QPRINT4)  JOBSEP(4)  TEXT('Default queue
  for all files using 4-part forms')
```

This command changes the number of job separators and the text that describes the output queue named QPRINT4. Four job separator pages are inserted before the spooled files for each job produced from the QPRINT4 output queue.

CHGPDGPRF (Change Print Descriptor Group Profile) Command



Notes:

K All parameters preceding this point are key parameters.

P All parameters preceding this point can be specified in positional form.

Purpose

The Change Print Descriptor Group Profile (CHGPDGPRF) command changes a print descriptor name for a particular user by altering the print descriptor values in the system.

Restriction: The user must have *OBJMGT authority to use this command.

Required Parameters

USER

Specifies the name of the user whose print descriptor group (PDG) profile is being changed.

***CURRENT:** The user profile under which the current job is running is used.

user-name: Specify the name of the user whose PDG profile is being changed.

PDG

Specifies the qualified name of the print descriptor group to be changed.

The name of the print descriptor can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

print-descriptor-group-name: Specify the name of the print descriptor group being changed.

Other Single Value

***SAME:** The value does not change.

PRTD

Specifies the name of the print descriptor within the print descriptor group that is used for the specified user profile.

***SAME:** The value does not change.

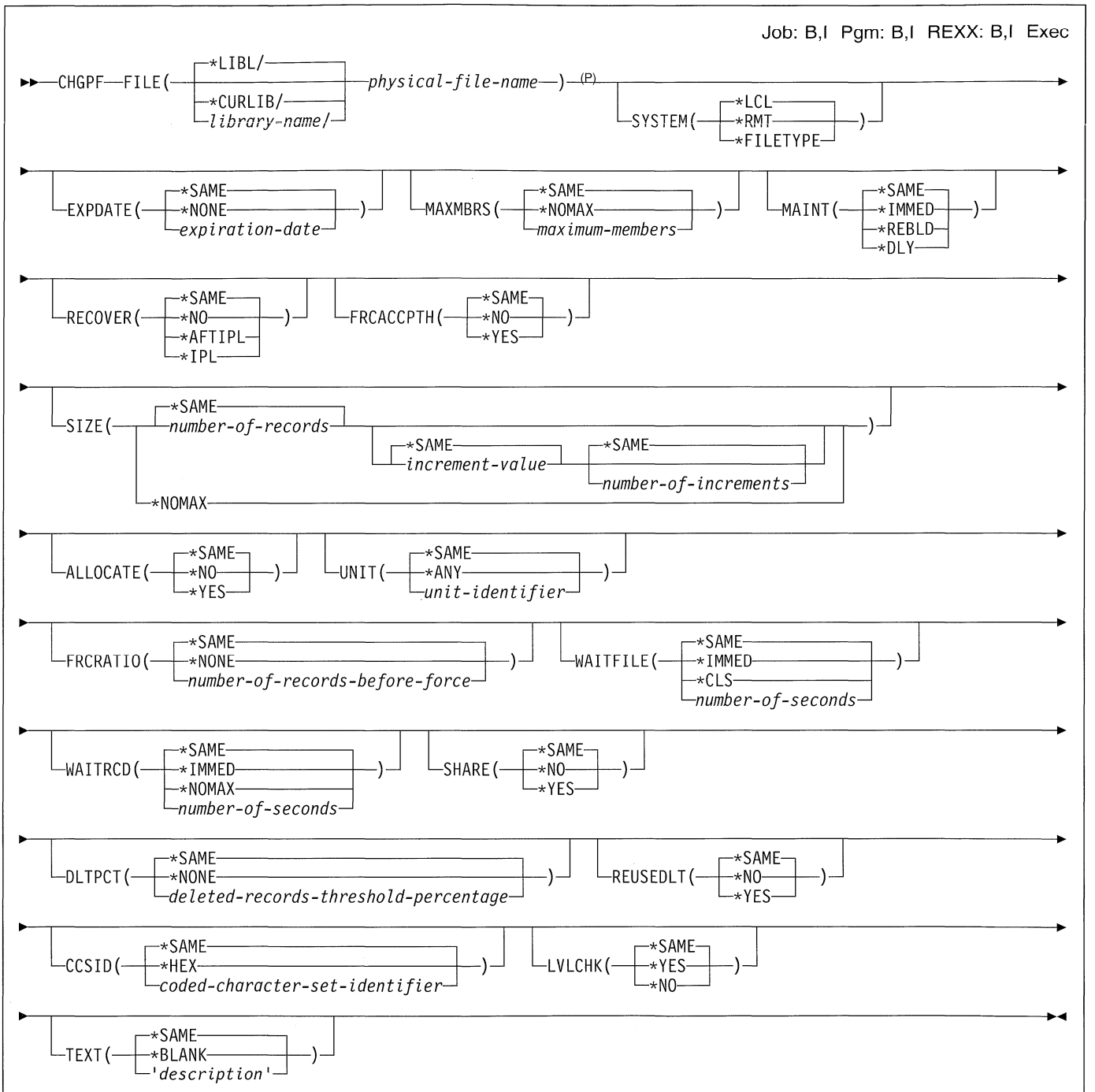
print-descriptor-name: Specify up to 256 characters for the name of the print descriptor.

Example

```
CHGPDGPRF   USER(JPSMITH)  PDG(*LIBL/LETTERS)
  PRTD(DRAFT_QUALITY)
```

This command changes the print descriptor for user profile JPSMITH. The print descriptor is changed to DRAFT_QUALITY in print descriptor group LETTERS.

CHGPF (Change Physical File) Command



Note:

P All parameters preceding this point can be specified in positional form.

Purpose

The Change Physical File (CHGPF) command changes the attributes of a physical file and all its members. The changed attributes are also used for all members subsequently added to the file. To change the attributes of a specific member, use the Change Physical File Member (CHGPFM) command.

Restriction: The user of this command must have object management authority and object operational authority for the file and read authority to the library. An exclusive-no-read lock is required, which means no one can be using the file for any purpose.

If a request to make an existing file re-use deleted records is made, but there are logical files over the physical file that specify "FIFO" or "LIFO" ordering for duplicate keys, the change is not allowed.

Required Parameter

FILE

Specifies the qualified name of the physical file to be changed.

Note: If a distributed data management (DDM) file is specified, then the physical file specified on the RMTFILE parameter of the Create Distributed Data Management File (CRTDDMF) command is changed on the remote system specified on the RMTLOCNAME parameter on the CRTDDMF command. More information is outlined in the SYSTEM parameter of this command.

The name of the file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

physical-file-name: Specify the name of the physical file.

Optional Parameters

SYSTEM

Specifies whether the physical file is changed on the local system or on a remote system.

***LCL:** The physical file is changed on the local system.

***RMT:** The physical file is changed on a remote system using DDM. The physical file name specified on the FILE parameter must be the name of the DDM file (created by using the Create Distributed Data Management File (CRTDDMF) command). The DDM file contains the name of the physical file to be changed (RMTFILE parameter on the CRTDDMF command) and the name of the remote system (RMTLOCNAME parameter on the CRTDDMF command) on which the file is to be changed.

***FILETYPE:** If the name specified on the FILE parameter is a DDM file, the physical file is changed on the remote system specified by the RMTLOCNAME parameter of that DDM file. Otherwise, the name specified on the FILE parameter to be changed must be the name of a local physical file.

EXPDATE

Specifies the expiration date. The files cannot be overwritten until the expiration date. The expiration date must be later than or equal to the current date.

Note: An attempt to open a file member that has exceeded its expiration date causes an error message to be sent to the user. (The Remove Member (RMVM) command is used to remove the member.) If EXPDATE is specified, all members in the file are changed. The expiration date must be later than or equal to the current date. The date must be specified in the format defined by the job attributes, DATFMT and DATSEP. The date must be enclosed in apostrophes if special characters are used in the format.

***SAME:** The value does not change.

***NONE:** No expiration date is specified.

expiration-date: Specify the date after which the member is not used.

MAXMBRS

Specifies the maximum number of members that the physical file can have at any time. The maximum number of members specified must be greater than or equal to the current number of members in the file.

***SAME:** The value does not change.

***NOMAX:** The system maximum is used.

maximum-members: Specify the maximum number of members that the physical file can have. Valid values range from 1 through 32767.

MAINT

Specifies the type of access path maintenance used for all members of the physical file. This parameter is valid only if the file has a keyed access path.

***SAME:** The value does not change.

***IMMED:** The access path is maintained for each physical file member whether the source physical file is opened or closed. The access path is changed whenever a record is updated, added to, or deleted from a member of this file or a logical file member based on a member of this file.

***REBLD:** The access path is rebuilt when a file member is opened. The access path is continuously maintained

until the member is closed; then the access path maintenance is ended. *REBLD is not valid for access paths that require unique key values.

***DLY:** The maintenance of the access path is delayed until the member is opened for use. The access path is changed only for records that are added, deleted, or changed since the file was last closed. (While the file is open, all changes made to based-on members are immediately reflected in the access paths of the members of the opened files, no matter what is specified for the MAINT parameter.) To prevent a lengthy rebuild time when the file is opened, *DLY should be specified only when the number of changes to the access path between a close operation and the next open operation are small (when key fields in records for this access path change infrequently). *DLY is not valid for access paths that require unique key values.

If the number of changes between a close operation and the next open operation reaches approximately 10% of the access path size, the system stops saving changes and the access path is completely rebuilt the next time the file is opened.

RECOVER

Specifies, for files having immediate or delayed maintenance on their access paths, when recovery processing of the file is done if a system failure occurs while the access path is being changed.

An access path having immediate or delayed maintenance is rebuilt during IPL (before any user can run a job), after the IPL is completed (while other jobs are running), or when the file is next opened. While the access path is being rebuilt, the file must not be used by any job.

During the IPL, an Override Access Path Recovery display lists those access paths that must be recovered and what the RECOVER parameter value is for each. The user can override the RECOVER parameter value on this display. More information is available in the *Advanced Backup and Recovery Guide*.

An access path having rebuild maintenance is rebuilt the next time its file is opened, the time that it normally is rebuilt. This parameter is valid only for files with a keyed access path.

***SAME:** The value does not change.

***NO:** The access path of the file is not rebuilt. The file's access path, if not valid, is rebuilt when the file is next opened.

***AFTIPL:** The file has its access path rebuilt after the IPL operation is completed. This option allows other jobs not using this file to begin processing immediately after the IPL has been completed. If a job tries to open the file while its access path is being rebuilt, a file open exception occurs.

***IPL:** The file has its access path rebuilt during the IPL. This ensures that the file's access path is rebuilt before the first user program tries to use it; however, no jobs start running until after all files that specify RECOVER(*IPL) have their access paths rebuilt.

FRCACCPTH

Specifies, for files with keyed access paths only, whether access path changes are forced to auxiliary storage along with the associated records in the file whenever the access path is changed. FRCACCPTH(*YES) minimizes (but does not remove) the possibility that an abnormal job end can cause damage to the access path, which then requires it to be rebuilt.

***SAME:** The value does not change.

***NO:** The access path and changed records are not forced to auxiliary storage whenever the access path is changed.

***YES:** The access path and changed records are forced to auxiliary storage whenever the access path is changed. If this value is specified, MAINT(*REBLD) cannot be specified.

FRCACCPTH(*YES) slows the response time of the system if the access path is changed in an interactive job. If the access path is changed frequently, the entire performance of the system is affected.

SIZE

Specifies the initial number of records in each member of the file, the number of records in each increment that is automatically added to the member size, and the number of times the increment is automatically applied.

A change to the initial number of records takes effect when a new member is added to the file or when a current member is cleared, restored, or reorganized. A change to the number of records to add for each increment, and the number of times the increment is automatically applied take effect the next time a member of the file needs an increment.

The total size of the member (first number of records plus the number of records added per increment times the number of increments) must be larger than the current size of the member. If it is smaller than the current size of the member, an error message is sent to the user, and the size does not change.

Element 1: Number of Records

***SAME:** The value does not change.

number-of-records: Specify the number of records (ranging from 1 through 16777215) that can be inserted before an automatic extension occurs. If automatic extensions are not wanted, enter zeros for the second and third values in the list.

Element 2: Increment Value

***SAME:** The value does not change.

increment-value: Specify the number of additional records (ranging from 0 through 32767) which, if greater than 10% of the size of the member when the maximum

number of records is reached, are to be added to the member during an automatic extension.

If the number specified is not greater than 10% of the member size and not equal to zero, the member size is increased by 10%.

Specify 0 to prevent automatic extensions. This value must be 0 if the value for the number of increments is 0.

Element 3: Maximum Number of Increments

***SAME:** The value does not change.

number-of-increments: Specify the maximum number of increments (ranging from 0 through 32767) that can be automatically added to the member. Enter a 0 to prevent automatic extensions. If the increment value is 0, the number of increments must be 0.

Other Single Values

***NOMAX:** The system maximum is used. This option cannot be specified if ALLOCATE(*YES) is in effect.

ALLOCATE

Specifies whether storage space is allocated for the initial number of records (SIZE parameter) for each physical file member when it is added. This change takes effect the next time a new member is added to the file or when a current member is cleared, restored, or reorganized.

***SAME:** The value does not change.

***NO:** When a new member is added, or when an existing member is cleared, restored, or reorganized, the system determines whether additional space is needed and allocates that amount.

***YES:** The amount of storage space specified in the first value of the SIZE parameter is allocated each time a new member is added, or each time an existing member is cleared, restored, or reorganized. If that amount of storage space is not available, the member is not added, and a message is sent to the user. If this parameter value is used, SIZE(*NOMAX) must not be in effect.

UNIT

Specifies the unit identifier of the auxiliary storage unit where the system attempts to allocate storage space for the file, its members, and their associated access paths. If the unit identifier is changed, this change takes effect for any increments that are applied to existing members or for new members to the file.

The system attempts to put any new increments for a member on the specified unit. However, the space allocated before the change in the unit specification remains on the unit on which it was originally allocated. To have the system attempt to place an entire member on the specified unit, (1) save the member, (2) delete the member from the system, and (3) restore the member on the system.

The unit identifier is a number ranging from 1 through 255 that is assigned when a new disk device is configured. Display and change the configured disk devices by using the Work with Disk Devices display from the Start System Services Tool (STRSST) command. More information on SST is in the *Advanced Backup and Recovery Guide.*

***SAME:** The value does not change.

***ANY:** The storage space for the file and its members is allocated on any available auxiliary storage unit.

unit-identifier: Specify the storage unit where the system attempts to allocate the storage space for the file and for all its members.

FRCRATIO

Specifies the number of inserted, updated, or deleted records that are processed before they are forced to auxiliary (permanent) storage. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

If the physical file is being recorded in a journal, a larger force write ratio, or *NONE, should be specified. More information on journal management is in the *Advanced Backup and Recovery Guide.*

***SAME:** The value does not change.

***NONE:** There is no force write ratio; the system determines when the records are written to auxiliary storage.

number-of-records-before-force: Specify the number of inserted, updated, or deleted records that are processed before they are explicitly forced to auxiliary storage.

WAITFILE

Specifies the number of seconds that the program waits for the file resources and session resources to be allocated when the file is opened, or for the device or session resources to be allocated when an acquire operation is performed to the file. If those resources are not allocated within the specified wait time, an error message is sent to the program. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

Note: An immediate allocation of the device by the device resource is required when an acquire operation is performed to the file.

***SAME:** The value does not change.

***IMMED:** The program does not wait; when the file is opened, an immediate allocation of the file resources is required.

***CLS:** The job default wait time is used as the wait time for the file resources being allocated.

number-of-seconds: Specify the number of seconds that a program waits for the file resources to be allocated to the job. Valid values range from 1 through 32767 seconds.

WAITRCD

Specifies the number of seconds that a program waits for a record to be updated or deleted, or for a record

read in the commitment control environment with LCKLVL(*ALL) specified. More information on record locking is in the *Database Guide*. If the record is not allocated in the specified wait time, an error message is sent to the program.

***SAME:** The value does not change.

***IMMED:** The program does not wait; when a record is locked, an immediate allocation of the record is required.

***NOMAX:** The system maximum is used.

number-of-seconds: Specify the number of seconds that a program waits for the file resources to be allocated to the job. Valid values range from 1 through 32767 seconds.

SHARE

Specifies whether the open data path (ODP) for the physical file is shared with other programs in the routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

More information on shared database files is in the *Database Guide*.

***SAME:** The value does not change.

***NO:** The ODP created by the program with this attribute is not shared with other programs in the routing step. Every time a program opens the file with this attribute, a new ODP to the file is created and activated.

***YES:** The ODP created with this attribute is shared with each program in the routing step that also specifies SHARE(*YES) when it opens the file.

Note: When SHARE(*YES) is specified and control is passed to a program, a read operation in that program retrieves the next input record. A write operation produces the next output record.

DLTPCT

Specifies the maximum percentage of deleted records that any member in the physical file can have. The percentage is based on the number of deleted records compared with the total record count in a member. This change takes effect the next time the file is opened and closed.

***SAME:** The value does not change.

***NONE:** No percentage is specified; the number of deleted records in the file members is not checked when a member is closed.

deleted-records-threshold-percentage: Specify the largest percentage of deleted records, ranging from 1 through 100, that any member in the file can have. If a value is larger than this percentage, a message is sent to the system history log (QHST) whenever the file is closed.

REUSEDLT

Specifies whether the space made available by deleting data entries is reclaimed.

***SAME:** The value does not change.

***NO:** The file does not reclaim space made available by deleting data entries.

***YES:** The file reclaims space made available by deleting data entries. More information about the algorithm used to reclaim the deleted data is in the *Database Guide*.

CCSID

Specifies the coded character set identifier (CCSID) being used to describe character data in the fields of the file.

Note: The CCSID cannot be changed if:

- There are any logical files defined over the physical file.
- Any explicit field- or file-level CCSIDs are specified on the CCSID keyword in DDS for fields in the physical file, IDDU, or SQL.
- Another file shares the physical file's format.
- The physical file is a program-described file.
- The physical file's format contains a concatenated field.

***SAME:** The CCSID does not change.

***HEX:** The CCSID 65535 is used, which indicates that the character data in the fields is treated as bit data and is not converted.

coded-character-set-identifier: Specify the CCSID being used.

If a DBCS field is in the physical file, the CCSID specified must have a corresponding mixed CCSID. More information on valid CCSIDs is in the *National Language Support Planning Guide*.

LVLCHK

Specifies whether the record format level identifiers in the program are checked against those in the logical file when the file is opened. If so, the record format identifiers in the program must match those in the logical file. This value can be overridden by the Override with Database File (OVRDBF) command at run time.

***SAME:** The value does not change.

***YES:** The level identifiers of the record formats are checked when the file is opened. If the level identifiers do not match, an error message is sent to the program requesting the open operation, and the file is not opened.

***NO:** The level identifiers are not checked when the file is opened.

TEXT

Specifies text that briefly describes the physical file. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Examples

Example 1: Changing Expiration Date for All Members

```
CHGPF    FILE(QGPL/INV)  EXPDATE('10/31/89')
```

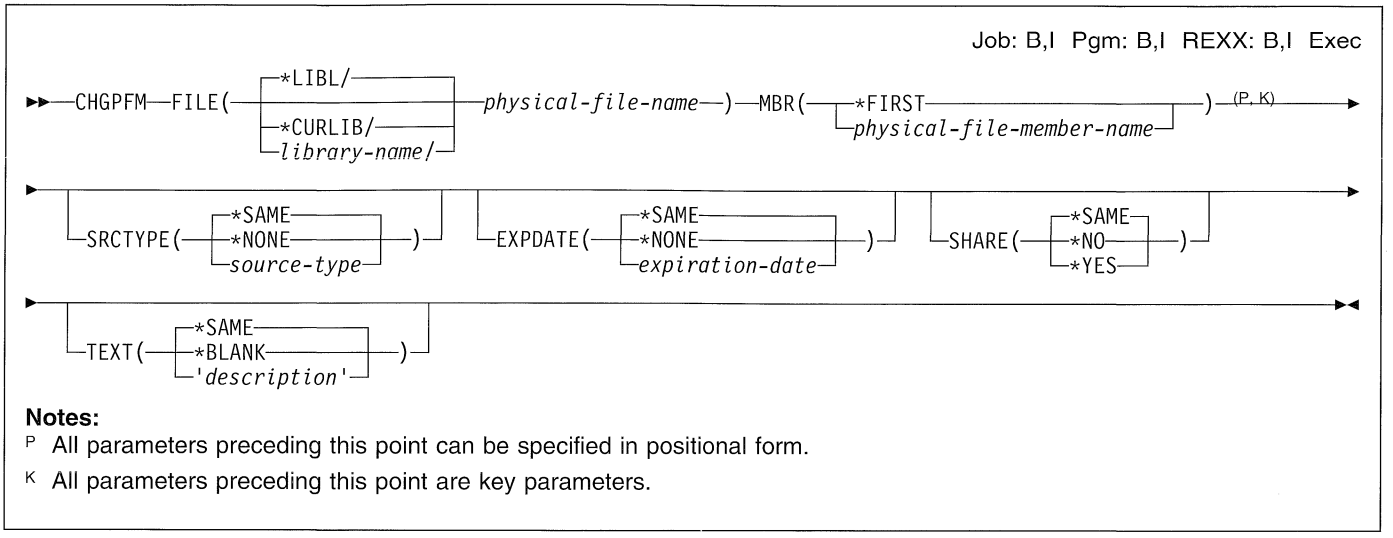
This command changes the expiration date for all members in physical file INV to October 31, 1989.

Example 2: Changing File Size

```
CHGPF  FILE(QGPL/DDMF)  SIZE(*NOMAX)  SYSTEM(*RMT)
```

This command changes the size of file INV located in the QGPL library on the remote system. Prior to specifying the above command, this user had created a DDM file by specifying the command CRTDDMF FILE(QGPL/DDMF) RMTFILE(QGPL/INV) RMTLOCNAME(AS400).

CHGPFM (Change Physical File Member) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Physical File Member (CHGPFM) command is used to change the attributes of a physical file member.

Restrictions:

1. To change a physical member, the user must have object management and object operational authority for the physical file that contains the member, and read authority to the file library.
2. No other user may be clearing or initializing the member, nor may any user be holding the file for exclusive use. Other users may have the member open, but the changes made to the member are not reflected in any open members. For the changes in open members to be effective, first close the member (this must be a full close if the member is open SHARE(*YES)) and then open it again.

Required Parameters

FILE

Specifies the qualified name of the physical file that contains the member to be changed.

The name of the physical file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

physical-file-name: Specify the name of the physical file.

MBR

Specifies the name of the member, or the first member (*FIRST) to be changed.

***FIRST:** The first member in the database file is used.

physical-file-member-name: Specify the name of the physical file member to be changed.

Optional Parameters

SRCTYPE

Specifies the source type of a member if this is a source file. The source type option is a character string of no more than 10 characters representing a name. The first character must be alphabetic (including the characters $, @, or #), and the remaining characters must be alphanumeric or an underline.

Notes:

1. The user of this command must assure the validity of the source type option.
2. The source type option can only be used with this command to change the source type attribute of a source file member.

***SAME:** The value does not change.

***NONE:** No source type is specified.

source-type: Specify the source type of a member.

EXPDATE

Specifies the expiration date. The files cannot be overwritten until the expiration date. The expiration date must be later than or equal to the current date.

Note: An attempt to open a file member that has exceeded its expiration date causes an error message to be sent. A member that has exceeded its expiration date may be changed to not exceed its expiration date by changing the

EXPDATE parameter. The expiration date must be later than or equal to the current day's date.

***SAME:** The value does not change.

***NONE:** No expiration date is specified.

expiration-date: Specify the date after which the member cannot be used. The date is specified in the format defined by the job attributes, DATFMT and DATSEP. The date is enclosed in apostrophes if special characters are used in the format.

SHARE

Specifies whether the open data path (ODP) for the physical file member is shared with other programs in the routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

More information on shared database files is in the *Database Guide*.

***SAME:** The value does not change.

***NO:** The ODP created by the program with this attribute is not shared with other programs in the routing step. Every time a program opens the file with this attribute, a new ODP to the file is created and activated.

***YES:** The ODP created with this attribute is shared with each program in the routing step that also specifies SHARE(*YES) when it opens the file.

Note: When SHARE(*YES) is specified and control is passed to a program, a read operation in that program retrieves the next input record. A write operation produces the next output record.

TEXT

Specifies text that briefly describes the physical file member. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

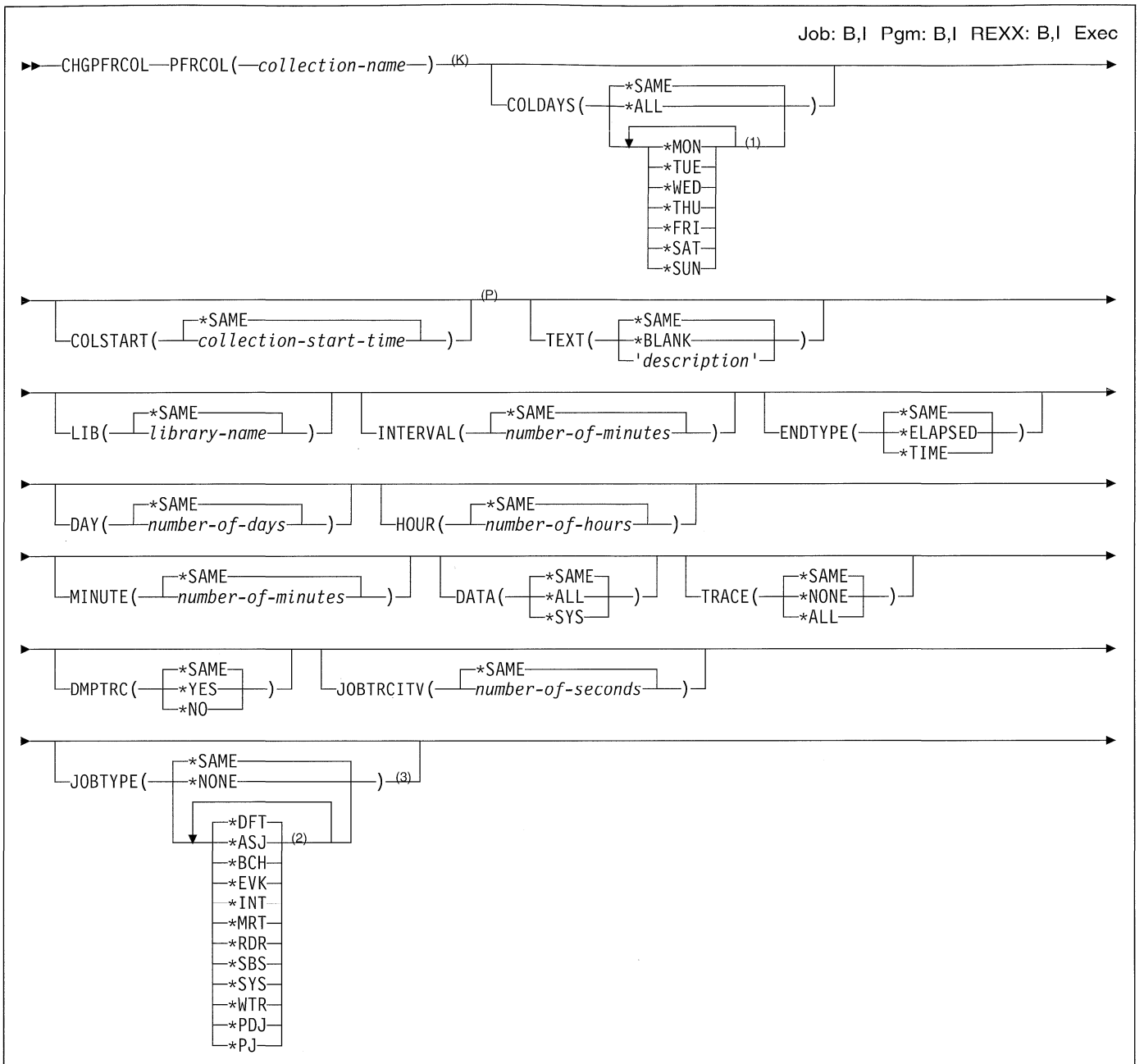
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGPFM   FILE(QGPL/INV)  MBR(FEB)  EXPDATE('10/31/90')
```

This command changes the member named FEB in the physical file INV in the QGPL library so that the expiration date of the member is now October 31, 1990.

CHGPFRCOL (Change Performance Collection) Command



Notes:

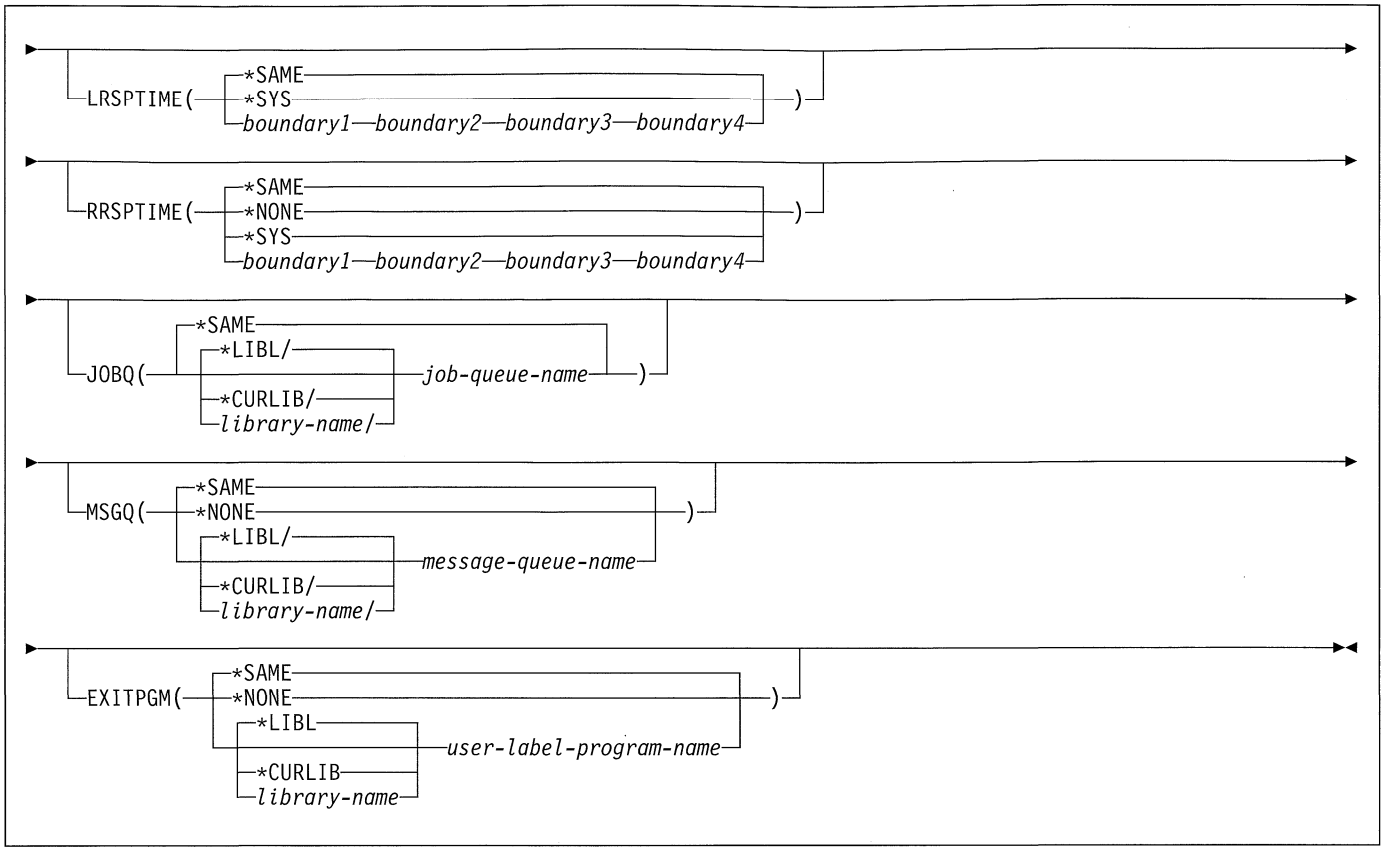
K All parameters preceding this point are key parameters.

1 A maximum of 7 repetitions

P All parameters preceding this point can be specified in positional form.

2 A maximum of 11 repetitions

3 A maximum of 10 unique job types can be traced.



Purpose

The Change Performance Collection (CHGPFRCOL) command allows the user to update the performance data collection. This is a weekly schedule that specifies when the Start Performance Monitor (STRPFRMON) command is sent. The collection consists of performance monitor information resulting from the weekly schedule specifying when the STRPFRMON command is to be sent.

This command causes the STRPFRMON command to be sent on the days specified on the COLDAYS parameter and at the time specified on the COLSTART parameter.

Required Parameter

PFRCOL

Specifies the name assigned to the performance data collection.

Optional Parameters

COLDAYS

Specifies the days of the week on which the STRPFRMON command is sent.

***SAME:** The value does not change.

***ALL:** The command is sent every day.

Days of the Week:

***MON:** The command is sent every Monday.

***TUE:** The command is sent every Tuesday.

***WED:** The command is sent every Wednesday.

***THU:** The command is sent every Thursday.

***FRI:** The command is sent every Friday.

***SAT:** The command is sent every Saturday.

***SUN:** The command is sent every Sunday.

COLSTART

Specifies the time of day when the STRPFRMON command is to be sent.

***SAME:** The value does not change.

collection-start-time: Specify the data collection start time. The time is specified in 24-hour format with or without a time separator as follows:

- With a time separator, specify a string of 5 or 8 digits where the time separator separates the hours, minutes, and seconds. If this command is entered from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command fails.
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds. Valid values for **hh**

range from 00 through 23. Valid values for **mm** and **ss** range from 00 through 59.

TEXT

Specifies text that briefly describes the performance data collection. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

LIB

Specifies the library where the database files for the performance monitoring information are collected. Each file that is not found in the specified library is automatically created by the system in the library.

***SAME:** The value does not change.

library-name: Specify the name of the library to be searched.

INTERVAL

Specifies the time interval (in minutes) between each collection of system performance monitoring information.

***SAME:** The value does not change.

number-of-minutes: Specify a collection interval ranging from 5 through 60 minutes.

ENDTYPE

Specifies how the performance monitor determines when to stop collecting information.

***SAME:** The value does not change.

***ELAPSED:** Collection stops after the specified amount of time has elapsed as specified on the HOUR and MINUTE parameters. Changes to the system clock do not affect the stop time once the performance monitor has been started.

***TIME:** Collection stops on the day and time specified on the DAY, HOUR, and MINUTE parameters.

DAY

Specifies, if ENDTYPE(*ELAPSED) is specified, the number of days to collect performance monitoring information. Specifies, if ENDTYPE(*TIME) is specified, the number of days to collect performance monitoring information based on the start day until the collection ends.

***SAME:** The value does not change.

number-of-days: Specify the number of days to collect performance monitoring information. Valid values range from 0 through 7.

HOUR

Specifies, if ENDTYPE(*ELAPSED) is specified, the number of hours to collect information. Specifies, if ENDTYPE(*TIME) is specified, the hour of the day when the collection ends.

***SAME:** The value does not change.

number-of-hours: Specify the number of hours to collect information. If ENDTYPE(*ELAPSED) is specified, valid values range from 0 through 168. If ENDTYPE(*TIME) is specified, valid values range from 0 through 23.

MINUTE

Specifies, if ENDTYPE(*ELAPSED) is specified, the number of minutes during which performance monitoring information is collected. Specifies, if ENDTYPE(*TIME) is specified, the minute of the specified hour in which the collection of performance monitoring information ends.

***SAME:** The value does not change.

number-of-minutes: Specify the number of minutes. Valid values range from 0 through 59.

DATA

Specifies the type of information collected.

***SAME:** The value does not change.

***ALL:** All information, including system information and communications information, is collected.

***SYS:** Only system information is collected.

TRACE

Specifies the type of internal trace being started.

***SAME:** The value does not change.

***NONE:** No trace is started.

***ALL:** All internal traces that contain performance related information are started.

DMPTRC

Specifies whether the trace is dumped when the collection of performance monitoring information ends.

***SAME:** The value does not change.

***YES:** The trace is dumped when the collection ends.

***NO:** The trace is not dumped when the collection ends. The trace can be dumped at a later time by using the Dump Trace (DMPTRC) command.

JOBTRCITV

Specifies the time (in CPU seconds) between each collection of the job trace information.

***SAME:** The value does not change.

number-of-seconds: Specify a time slice quantum value (in seconds) ranging from 0.5 through 9.9.

JOBTYPE

Specifies the type of job being traced. A maximum of 10 unique job types in addition to *DFT can be traced.

***SAME:** The value does not change.

***DFT:** Batch and autostart jobs are traced.

***ASJ:** Autostart jobs are traced.

***BCH:** Batch jobs are traced.

***EVK:** Jobs started by a procedure start request are traced.

***INT:** Interactive job types are traced.

***MRT:** Multiple requester terminal jobs are traced.

***RDR:** Reader jobs are traced.

***SBS:** Subsystem monitor jobs are traced.

***SYS:** System jobs are traced.

***WTR:** Writer jobs are traced.

***PDJ:** Print driver jobs are traced.

***PJ:** Prestart jobs are traced.

Other single values

***NONE:** No jobs are traced.

LRSPTIME

Specifies the local work station response time categories. The performance monitor keeps track of interactive response times for each local work station attached to a controller that supports the collection of response time information. The response times are grouped into five categories; this parameter defines each category. Each value must contain a three-position decimal number with one decimal position.

***SAME:** The value does not change.

***SYS:** The system response categories are:

- 0-1 seconds
- 1-2 seconds
- 2-4 seconds
- 4-8 seconds
- Longer than 8 seconds

Element 1: The First Boundary

boundary1: Specify the first response time boundary. All responses falling between zero and this boundary value are counted in the first response time category. This is a DEC(3,1) variable.

Element 2: The Second Boundary

boundary2: Specify the second response time boundary. All responses falling between the first boundary value and this boundary value are counted in the second response time category. This is a DEC(3,1) variable.

Element 3: The Third Boundary

boundary3: Specify the third response time boundary. All responses falling between the second boundary value and this boundary value are counted in the third response time category. This is a DEC(3,1) variable.

Element 4: The Fourth Boundary

boundary4: Specify the fourth response time boundary. All responses falling between the third boundary value and this boundary value are counted in the fourth response time category. This is a DEC(3,1) variable. All response times greater than this value are counted in the fifth response time category.

RRSPTIME

Specifies the remote work station response time categories. The performance monitor keeps track of interactive response times for each remote work station attached to a controller that supports the collection of response time information. The response times are grouped into five categories; this parameter defines each category. Each value must contain a three-position decimal number, with one decimal position.

***SAME:** The value does not change.

***NONE:** Remote work station response time is not collected.

***SYS:** The system response categories are:

- 0-1 seconds
- 1-2 seconds
- 2-4 seconds
- 4-8 seconds
- Longer than 8 seconds

Element 1: First Boundary

boundary1: Specify the first response time boundary. All responses falling between zero and this boundary value are counted in the first response time category. This is a DEC(3,1) variable.

Element 2: Second Boundary

boundary2: Specify the second response time boundary. All responses falling between the first boundary value and this boundary value are counted in the second response time category. This is a DEC(3,1) variable.

Element 3: Third Boundary

boundary3: Specify the third response time boundary. All responses falling between the second boundary value and this boundary value are counted in the third response time category. This is a DEC(3,1) variable.

Element 4: Fourth Boundary

boundary4: Specify the fourth response time boundary. All responses falling between the third boundary value and this boundary value are counted in the fourth response time category. This is a DEC(3,1) variable. All response times greater than this value are counted in the fifth response time category.

JOBQ

Specifies the qualified name of the job queue on which this job is placed.

***SAME:** The value does not change.

The name of the job queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

job-queue-name: Specify the name of the job queue.

MSGQ

Specifies the qualified name of the message queue to which messages are sent.

Note: The performance monitor can send its messages to the user-defined message queue and the system operator message queue (QSYS/QSYSOPR). Messages sent to the user-defined message queue communicates status information relating to the performance monitoring of start-up and end operations, as well as general system problems.

***SAME:** The value does not change.

***NONE:** Messages are sent to the system operator message queue only.

The name of the message queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

message-queue-name: Specify the name of the message queue that is used to receive messages.

EXITPGM

Specifies the user-written exit program that is called to process the performance data collected by this command.

***SAME:** The value does not change.

***NONE:** No exit program is specified.

The name of the exit program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

user-label-program-name: Specify the user label name of the exit program to be used.

Examples

Example 1: Changing Collection Date, Time and Text

```
CHGPFRCOL  PFRCOL(MONDAYS)  COLDAYS(*MON)
  COLSTART(0800)  TEXT('Mondays 8:00 to 10:00')
```

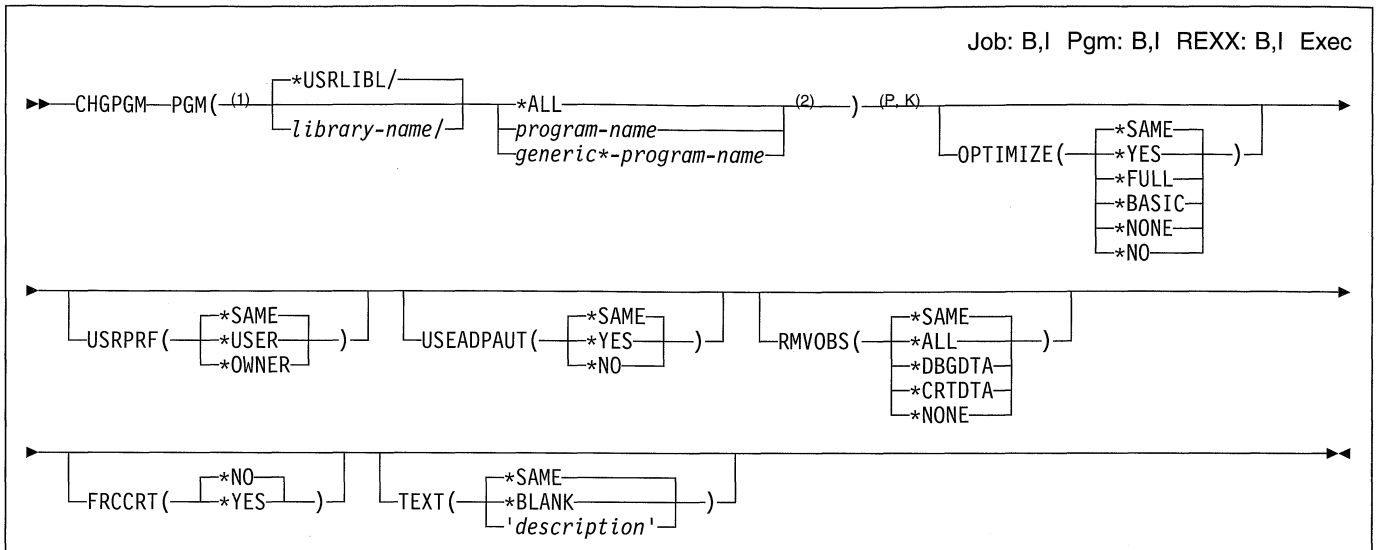
This command changes the collection named MONDAYS. The Start Performance Monitor (STRPFRMON) command is sent every Monday at 8:00 a.m.

Example 2: Changing Collection Date and Time

```
CHGPFRCOL  PFRCOL(AFTERNOON)
  COLDAYS(*MON *TUE *WED *THU *FRI)
  COLSTART(1330)  ENDTYPE(*TIME)
  HOUR(16)  MINUTE(30)
```

This command changes the collection named AFTERNOON. The Start Performance Monitor (STRPFRMON) command is sent every day, Monday through Friday, at 1:30 p.m.

CHGPGM (Change Program) Command



Notes:

1 QSYS and QGDDM cannot be specified for the library unless the only change is a removal of observability.

2 *ALL or a generic request cannot be specified with *USRLIBL for the library qualifier.

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Program (CHGPGM) command changes the attributes of a program without requiring that it be recompiled. The attributes that can be changed are the optimization attribute, the user profile attribute, the use-adopted-authority attribute, and the program text. The user can also force re-creation of a program even if the attributes being specified are the same as the current attributes.

Restrictions:

1. The user must have delete authority for the library for the program that is being changed.
2. The user must have object management authority for the program that is being changed.
3. The user must have *USE, *DLT, and *ADD authority for the library to change the optimization attribute (OPTIMIZE) or to force program re-creation by specifying FRCCRT(*YES).
4. Only the program owner, or a user with QSECOFR authority, can change the user profile attribute (USRPRF). Programs in library QSYS and QGDDM cannot be changed.
5. Before using this command, the user must have *SECADM and *ALLOBJ authority.

If the program being changed is not an Integrated Language Environment* (ILE*) program, other jobs running the program fail when the program optimization attribute (OPTIMIZE) or user profile attribute (USRPRF) are changed, or program re-creation is forced by specifying FRCCRT(*YES). If the program being changed is an ILE program, other jobs running the program do not fail.

Required Parameter

PGM

Specifies the programs whose attributes are being changed. *USRLIBL cannot be specified or defaulted for the library qualifier when a generic name or *ALL is specified for the program qualifier.

The name of the program can be qualified by one of the following library values:

*USRLIBL: Only the libraries in the user portion of the job's library list are searched.

library-name: Specify the name of the library to be searched.

***ALL:** All programs in the specified library to which the user has some authority (for example, *USE authority) are selected for change.

program-name: Specify the name of the programs whose attributes are being changed.

generic-program-name:* Specify the generic name of the program. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the

generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be changed only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

Optional Parameters

OPTIMIZE

Specifies whether the program is optimized. This parameter removes redundant instructions from the specified programs. Changing the current optimization level of an Integrated Language Environment (ILE) program causes the system to re-create an ILE program with the new optimization level.

***SAME:** The value does not change.

***YES:** The program is optimized. In most cases, optimized programs make more efficient use of system resources. When debugging ILE programs at this level, variables can be displayed, but the displayed value may not be the current value. Variables can also be changed but using the variables changed at this level may cause unexpected results.

***FULL:** This value is identical to *YES.

***BASIC:** Some optimization is performed on ILE programs. When debugging ILE programs at this level, variables can be displayed, but the displayed value may not be the current value. Variables can also be changed but using the variables changed at this level may cause unexpected results. An informational message is sent and no optimization is performed when OPTIMIZE(*BASIC) is specified for original program model (OPM) programs.

***NO:** The program is not optimized. Variables can be displayed and changed when debugging ILE programs at this optimization level.

***NONE:** This value is identical to *NO.

USRPRF

Specifies whether the authority checking done while this program is running includes only the user who is running the program (*USER) or both the user running the program and the program owner (*OWNER). The profiles of the program user or both the program user and the program owner are used to control which objects can be used by the program, including the authority the program has for each object. Only the program owner or a user with QSECOFR authority can change the user profile attribute.

***SAME:** The value does not change.

***USER:** The program runs under the user profile of the program's user.

***OWNER:** The user profiles of both the program's owner and the program's user are used when the program is processed. The collective sets of object authority in both user profiles are used to find and access objects during program processing. Authority from the owning user profile's group profile is not included in the authority for the running program.

USEADPAUT

Specifies whether program adopted authority from previous programs in the call stack will be used as a source of authority when this program is running.

***SAME:** The value does not change.

***YES:** Program adopted authority from previous invocation levels is used when this program is running.

***NO:** Program adopted authority from previous invocation levels is not used when this program is running.

RMVOBS

Specifies whether the observable information associated with programs is removed.

***SAME:** The value does not change.

***ALL:** All of the observable information associated with the program is removed, if possible. If the program requires the observable information to ensure that it runs correctly, that information is not removed.

***DBGDTA:** Information is removed from an ILE program to allow the program to be debugged. An informational message is sent and no observability is removed when RMVOBS(*DBGDTA) is specified for original program model (OPM) programs.

***CRTDTA:** Information is removed from an ILE program to allow the program to be re-created or to change the optimization level. An informational message is sent and no observability is removed when RMVOBS(*CRTDTA) is specified for original program model (OPM) programs.

***NONE:** None of the observable information associated with the program is removed.

FRCCRT

Specifies whether program re-creation is forced.

***NO:** Program re-creation is not forced unless the optimization (OPTIMIZE), use adopted authority (USEADPAUT), or user profile (USRPRF) parameters are changed. This option allows the system to determine whether a change is required.

***YES:** Program re-creation is forced whether or not the optimization (OPTIMIZE), use adopted authority (USEADPAUT) or user profile (USRPRF) parameter has been changed.

TEXT

Specifies text that briefly describes the program. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Examples

Example 1: Optimizing a Program

```
CHGPGM   PGM(PROG1/SERVICE)  OPTIMIZE(*YES)
  USRPRF(*OWNER)
```

The program SERVICE in library PROG1 is optimized, and the user profile under which it is processed is changed to include the program owner's user profile. Only the owner of program PROG1/SERVICE, or a user with security officer authority, can change the USRPRF attribute. The program is re-created only if the attributes specified differ from those of the current program.

Example 2: Changing Text for a Program

```
CHGPGM   PGM(*USRLIBL/KNUTE)
  TEXT('Program description')
```

This command changes the text for program KNUTE. The user portion of the library list is used to find the program.

Example 3: Optimizing Multiple Programs

```
CHGPGM   PGM(PROG1/ACE*)  OPTIMIZE(*YES)
```

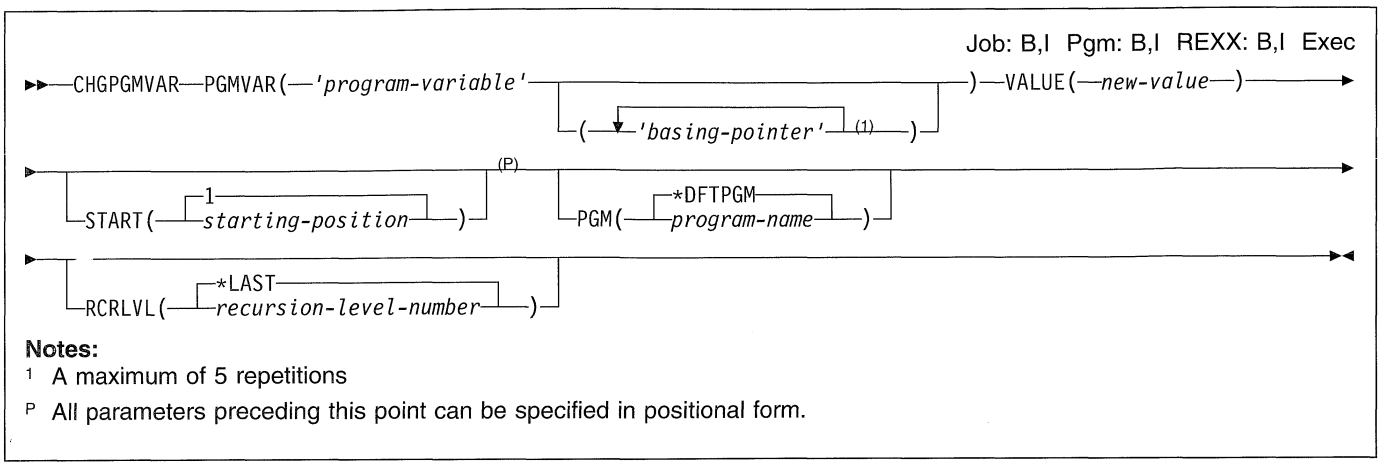
All programs in library PROG1 that have not been optimized, and whose names begin with ACE, are optimized.

Example 4: Changing Text of Multiple Programs

```
CHGPGM   PGM(PROG2/*ALL)  TEXT('Generic Text')
```

This command changes the text of all programs in library PROG2 to Generic Text.

CHGPGMVAR (Change Program Variable) Command



Notes:
1 A maximum of 5 repetitions
P All parameters preceding this point can be specified in positional form.

Purpose

The Change Program Variable (CHGPGMVAR) command changes the value of a variable in a program being debugged. Only string (character or bit) and numeric variables can be changed by using this command. A portion of a character string or bit string can be changed; the length of the data being changed is the length of the data specified in the VALUE parameter.

Note: Depending on where the variable is allocated, the duration of a change varies. For a variable in machine-interface (MI) static storage, the change lasts while the program is active. For an automatic variable, the change lasts until the call of the program is ended. Static variables are allocated either in MI static storage or in spaces controlled by language-dependent rules.

Restrictions:

1. This command is valid only in debug mode. To start debug mode, refer to the STRDBG (Start Debug) command.
2. This command cannot be used if the user is servicing another job, and that job is on a job queue, or is being held, suspended, or ended.
3. This command cannot be used to change variables in a bound program.
4. This command cannot be used to change variables that are write-protected or within the system domain, unless the user has *SERVICE special authority.

Required Parameters

PGMVAR
Specifies the name of the program variable being changed in a high-level language (HLL) program or MI program.

Element 1: Program Variables

'program-variable': Specify the name of the program variables being changed. If the variable name contains special characters, it must be enclosed in apostrophes.

If the program variable is an array, the subscripts representing an element in the array can be specified. Up to 132 characters may be specified for this program variable entry. This includes any qualifiers, subscripts, embedded blanks, parentheses, and commas. It does not include the enclosing apostrophes when special characters are used. An integer, machine-interface object-definition-table-vector (MI ODV) number, or a numeric variable name can be specified for a subscript. For more information on the program-variable value, refer to Appendix C, "Parameter Values Used for Testing and Debugging."

Some examples are:

```
PGMVAR(A)
PGMVAR('A(2,B)')
PGMVAR('B(I1,*,I3)')
PGMVAR('VAR1 OF A(I,J IN B)')
```

Element 2: Basing Pointers

'basing-pointer': Specify the basing pointers for the variable to be recorded. In some languages, the program variable may be based on a pointer variable. This set of values allows the user to explicitly specify the basing pointers for the variable to be recorded. Each basing pointer name must be enclosed in apostrophes if it contains special characters.

If the basing pointer is an array, the subscripts representing an element in the array must be specified. Up to 132 characters can be specified for a basing pointer name. This includes any qualifiers, subscripts, embedded blanks, parentheses, and commas. It does not include the enclosing apostrophes when special characters are used. An integer, MI ODV number, or a numeric variable name can be specified for a subscript. For more information on the basing pointer value, refer

to Appendix C, "Parameter Values Used for Testing and Debugging."

Some examples are:

```
PGMVAR(('VAR1(B,5)' 'PTR2(C,P2)'))
PGMVAR((VAR2 (BASEPTRA BASEPTRB)))
```

VALUE

Specifies the new value for the program variable. Depending on the variable type, its value must be specified using the following rules:

- The value for a character variable must be enclosed in apostrophes if it contains blanks, special characters, or numeric characters (for example, `'ABC DE'`, which contains an embedded blank, or `'37.92'`, which contains only a decimal point and numeric characters).
- The value for a bit variable must be enclosed in apostrophes, and can contain only the characters 0 and 1. Each character specified sets one bit in the variable to the value specified. A hexadecimal value cannot be specified to change a bit variable.
- For a fixed length string, a null string value can only be specified for a string with a declared length of zero.
- For a varying-length string, a null string can be specified regardless of its declared length. Any time a varying-length string is changed, the current length of the string is also implicitly changed. When the VALUE string is not null, the current length is changed to a value that reflects the last byte affected by the change (equal to the value specified in the START parameter, plus the VALUE length, minus one). If the VALUE string is null, the current string length is changed to one less than the START parameter value.

  A varying-length string can be truncated or extended by specifying a combination of a START position and a VALUE whose length is less than or greater than the current string length. A null VALUE string can be specified to truncate a varying-length string to any length (including zero).
- The value for a numeric variable can be specified with or without a decimal point (comma or period) and with or without a plus or minus sign. If a negative value is specified, it must include a minus (-) sign. If a decimal point is not entered in the value, it is assumed to be on the right of the last digit entered; that is, the value is assumed to be an integer (whole number) only. The user cannot change a fixed-point numeric variable with a value that has more nonzero integers or fractional digits than the variable was declared with. In addition, the user cannot specify more than 15 integer digits and 9 fractional digits for the new value.

  If, for example, a numeric variable is defined as a five-position decimal value of which two positions are the fraction portion, the following values can be specified:

Specified Value	Assumed Value
2.7 or 2,7	2.70
27 or 27.00	27.00
-27	-27.00

- The value for any numeric variable can be specified using one of the following:
  - A decimal value (includes integer numbers)
  - A floating-point constant
- A floating-point variable can be changed to positive or negative infinity or not a number by using the special values of:

  *INF (positive infinity)
  *NEGINF (negative infinity)
  *NAN (not-a-number)
- Values for all variable types except bit can be entered in hexadecimal form (X'058F' for packed decimal 58).

Optional Parameters

START

Specifies, for string variables only, the starting position in the string from which its value is to change. For a bit string, the value specifies the starting bit position; for a character string, the value specifies the starting character position.

1: The first position of the program variable is the starting position in the string to change.

starting-position: Specify the position number in the program variable that specifies the first position to change in the string. The starting-position number must not be greater than the length of the string, except that START(1) is allowed if the maximum length is zero. The length of the change value specified on the VALUE parameter, plus the START position minus one, must not be greater than the maximum string length.

Varying length strings have rules affecting the validity and meaning of the START parameter. Refer to the appropriate high-level language reference manual for specific information.

PGM

Specifies the name of the program that contains the program variable whose value is to be changed.

***DFTPGM:** The program currently specified as the default program contains the variable to be changed.

program-name: Specify the name of the program that contains the variable to be changed. The same name must already have been specified on the Start Debug (STRDBG) command or Add Program (ADDPGM) command.

RCRLVL

Specifies which recursion level of the program contains the variable whose value is being changed. Changes made to static variables automatically affect all recursion levels. Recursion level 1 is the first (or earliest) call of the program, recursion level 2 is the second call of the program, and so on to the last (most recent) recursion level in the stack. For example, if program A calls program B, and then program B calls program A, a new recursion level of program A is formed. If the first recursion level of program A contains the variable being changed, RCRLVL(1) must be specified. Some high-level languages also allow recursive procedures. For these programs, refer to the appropriate high-level language manual for more information.

***LAST:** The last (most recent) call of the specified program contains the variable being changed.

recursion-level-number: Specify the recursion level of the program that contains the variable being changed.

Examples

Example 1: Changing the Program Variable

```
DCLVAR(&AMT)  TYPE(*DEC)  LEN(5  2)
     •
     •
CHGPGMVAR   PGMVAR('&AMT')  VALUE(16.2)
```

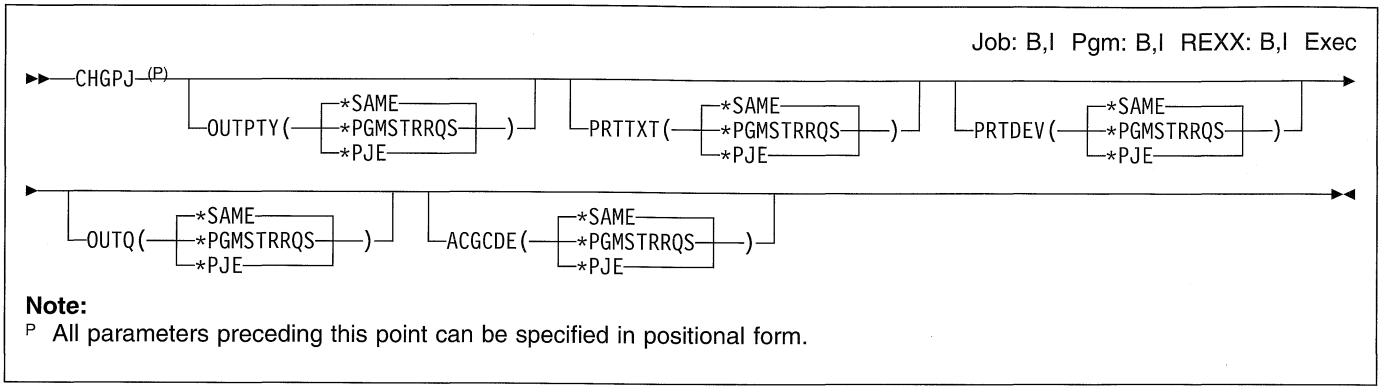
The first command, which is used in a CL program, declares the CL variable &AMT as a five-position decimal value having a 3-digit integer and a 2-digit fraction. The Change Program Variable (CHGPGMVAR) command is used to change the value of &AMT to 16.20. If VALUE is coded as 16 or 16.00, the value accepted is 16.00; if -16 is coded, the value accepted is -16.00. However, if 1600 is coded, an error occurs because the system assumes that, if no decimal point is coded, it is always on the right of the last digit coded.

Example 2: Changing a Value in a Specific Position

```
CHGPGMVAR   PGMVAR(PARTNO)  VALUE('56')  START(4)
```

This command changes, starting in position 4, the program variable PARTNO to 56.

CHGPJ (Change Prestart Job) Command



Note:

P All parameters preceding this point can be specified in positional form.

Purpose

The Change Prestart Job (CHGPJ) command changes the job attributes of the prestart job to those in the job description specified in the user profile associated with the program start request, or to those in the job description specified in the prestart job entry.

Restrictions:

1. This command is restricted to a user with *USE authority.
2. The user must have *USE authority to the job description. To change the OUTQ, the job must have authority to the new output queue.

Optional Parameters

OUTPTY

Specifies the output priority for spooled files produced by this job.

***SAME:** The value does not change.

***PGMSTRRQS:** The value specified in the job description for the user profile of the program start request is used.

***PJE:** The value specified in the job description for the prestart job entry in the subsystem description is used.

PRTTXT

Specifies up to 30 characters of text to be printed at the bottom of each page of output. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***PGMSTRRQS:** The value specified in the job description for the user profile of the program start request is used.

***PJE:** The value specified in the job description for the prestart job entry in the subsystem description is used.

PRTDEV

Specifies the qualified name of the default printer device for this job. If OUTQ(*DEV) is specified, the file is placed on an output queue with the same name as the printer.

***SAME:** The value does not change.

***PGMSTRRQS:** The value specified in the job description for the user profile of the program start request is used. If *USRPRF is specified for this parameter value in the job description, the value specified in the user profile of the program start request is used.

***PJE:** The value specified in the job description for the prestart job entry in the subsystem description is used. If *USRPRF is specified for this parameter value in the job description, the value specified in the user profile for the prestart job entry (USER parameter) is used.

OUTQ

Specifies the qualified name of the output queue.

***SAME:** The value does not change.

***PGMSTRRQS:** The value specified in the job description for the user profile of the program start request is used. If *USRPRF is specified for this parameter value in the job description, the value from the user profile of the program start request is used.

***PJE:** The value specified in the job description for the prestart job entry in the subsystem description is used. If *USRPRF is specified for this parameter value in the job description, the value from the user profile for the prestart job entry (USER parameter) is used.

ACGCDE

Specifies the accounting code used for the job.

***SAME:** The value does not change.

***PGMSTRRQS:** The value specified in the job description for the user profile of the program start request is used. If *USRPRF is specified for this parameter value in the job description, the value from the user profile of the program start request is used.

***PJE:** The value specified in the job description for the prestart job entry in the subsystem description is used.

If *USRPRF is specified for this parameter value in the job description, the value from the user profile for the prestart job entry (USER parameter) is used.

Examples

Example 1: Changing the Output Priority

```
CHGPJ  OUTPTY(*PGMSTRRQS)
```

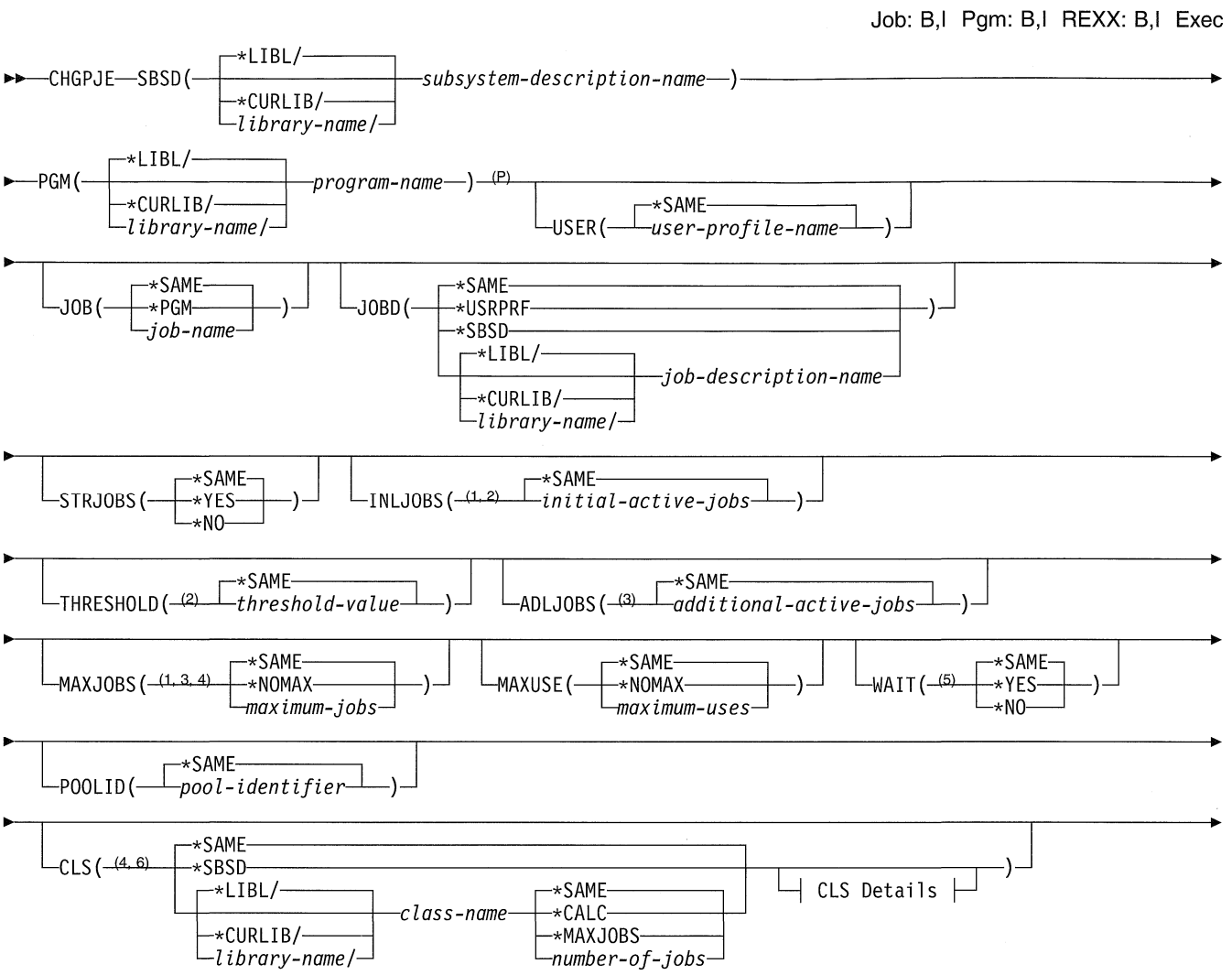
This command changes the output priority for spooled files produced by this job to the OUTPTY in the job description associated with the user profile of the program start request.

Example 2: Changing the Output Queue

```
CHGPJ  OUTQ(*PJE)
```

This command changes the output queue for spooled files produced by this job to the OUTPUT queue in the job description specified in the prestart job entry.

CHGPJE (Change Prestart Job Entry) Command



Notes:

1 The value of the INLJOBS parameter must be less than or equal to the value of the MAXJOBS parameter.

2 The value of the INLJOBS parameter must be greater than or equal to the value of the THRESHOLD parameter.

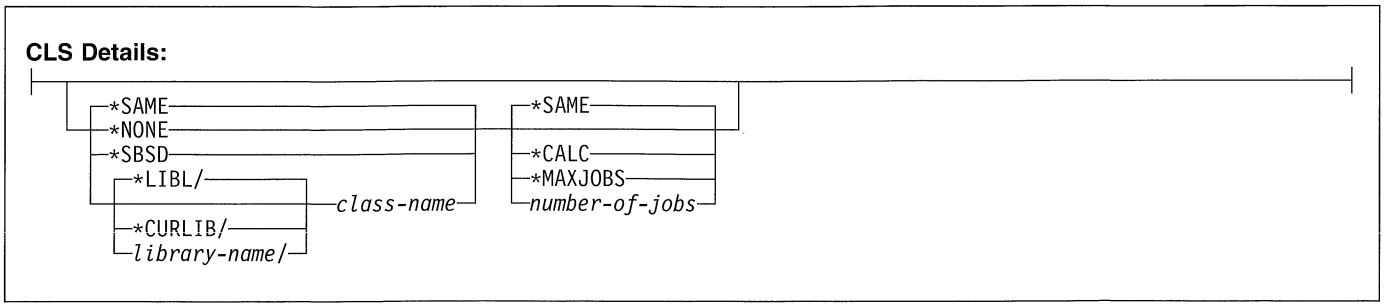
3 The value of the ADLJOBS parameter must be less than the value of the MAXJOBS parameter.

4 If the value for the MAXJOBS parameter is changed, the value specified for the CLS parameter might also need to be changed. If MAXJOBS(*CALC) is specified, the system recalculates the value for the number of jobs to use the specified class.

5 When using switched communications lines or BSC protocols, determine whether the lines will go into error recovery if WAIT(*YES) is specified.

P All parameters preceding this point can be specified in positional form.

6 Two classes may be specified for this parameter.



Purpose

The Change Prestart Job Entry (CHGPJE) command changes a prestart job entry in the specified subsystem description. The associated subsystem may be active when the prestart job entry is changed. Changes made to the entry when the subsystem is active are reflected over time. New prestart jobs created after the command is issued use the new job-related values. This command identifies prestart jobs that are started when the subsystem is started.

Restriction: This command is restricted to a user with *USE and object management authorities for the subsystem description and *USE authority for the user profile and the job description.

Required Parameters

SBSD

Specifies the qualified name of the subsystem description that contains the prestart job entry being changed.

The name of the subsystem description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

subsystem-description-name: Specify the name of the subsystem description that contains the prestart job entry.

PGM

Specifies the qualified name of the program that identifies the prestart job entry being changed. This program name is used to match an incoming program start request with an available prestart job. Two entries with the same program name can exist in a single subsystem description, but they must have different library names. If the program does not exist when the entry is added, a library qualifier must be specified because the qualified name is retained in the subsystem description.

The name of the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the program.

Optional Parameters

USER

Specifies the user profile under which the prestart job runs when it is not handling a program start request.

***SAME:** The value of this parameter does not change.

user-profile-name: Specify the name of the user profile used for the prestart job.

JOB

Specifies the name of the prestart job that is started.

***SAME:** The value does not change.

***PGM:** The job name is the same name as the program name, specified by the PGM parameter.

job-name: Specify the name of the prestart job.

JOBD

Specifies the name of the job description used. If the job description does not exist when the entry is changed, a library qualifier must be specified because the qualified job description name is retained in the subsystem description. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***USRPRF:** The job description name specified in the user profile, which was named in the USER parameter, is used.

***SBSD:** The job description having the same name as the subsystem description named in the SBSD parameter is used.

The name of the job description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

job-description-name: Specify the name of the job description being used for this prestart job. If no library is specified, the library list (*LIBL) of the job where this command is run is used to find the job description.

STRJOBS

Specifies whether prestart jobs are started when the subsystem is started.

***SAME:** The value does not change.

***YES:** The prestart jobs are started when the subsystem is started.

***NO:** The prestart jobs are not started at the time the subsystem is started. The Start Prestart Jobs (STRPJ) command must be used to start these prestart jobs.

INLJOBS

Specifies the initial number of prestart jobs that are started when the subsystem specified in the SBSD parameter is started.

Notes:

1. The value of this parameter must be less than or equal to the value of the MAXJOBS parameter.
2. The value of this parameter must be greater than or equal to the value of the THRESHOLD parameter.

***SAME:** The value does not change.

initial-active-jobs: Specify the number of prestart jobs that are started when the subsystem is started. Valid values range from 1 through 1000.

THRESHOLD

Specifies when additional prestart jobs are started. When the pool of available prestart jobs (jobs available to service program start requests) is reduced below this number, more jobs (specified by the ADLJOBS parameter) are started and added to the available pool.

Note: The value of this parameter must be less than or equal to the value of the INLJOBS parameter.

***SAME:** The value does not change.

threshold-value: Specify the minimum number of prestart jobs that must be available before additional prestart jobs are started. Valid values range from 1 through 1000.

ADLJOBS

Specifies the additional number of prestart jobs that are started when the number of prestart jobs drops below the THRESHOLD parameter.

Note: The value of this parameter must be less than the value of the MAXJOBS parameter.

***SAME:** The value does not change.

additional-active-jobs: Specify the number of additional prestart jobs to start. Valid values range from 0 through 1000.

MAXJOBS

Specifies the maximum number of prestart jobs that can be active at the same time for this prestart job entry.

Notes:

1. The value of this parameter must be greater than or equal to the value of the INLJOBS parameter.
2. The value of this parameter must be greater than the value of the ADLJOBS parameter.
3. If the value specified for this parameter is changed, the value specified for the CLS parameter might also need to be changed.

***SAME:** The value does not change.

***NOMAX:** There is no maximum number of jobs that can be active at the same time.

maximum-jobs: Specify the maximum number of prestart jobs that can be active at the same time. Valid values range from 1 through 1000.

MAXUSE

Specifies the maximum number of program start requests that can be handled by each prestart job before the subsystem ends the job in a controlled manner by issuing an ENDJOB command with a value of *CNTRLD on the OPTION parameter.

***SAME:** The value does not change.

Note: If *NOMAX is specified, the prestart jobs may end abnormally because the job has exceeded the allowed maximum job log size, the maximum number of spool files, the maximum CPU time, or the maximum temporary storage space required.

***NOMAX:** There is no maximum number of jobs that can be active at the same time.

maximum-uses: Specify the maximum number of program start requests that a prestart job can handle before it is ended. Valid values range from 1 through 1000.

WAIT

Specifies whether program start requests wait for a prestart job to become available or are rejected if a prestart job is not immediately available when the program start request is received.

Note: See the *CL Programmer's Guide* for the time-out considerations for each communications type.

***SAME:** The value does not change.

***YES:** Program start requests wait until there is either an available prestart job, or until a prestart job is started, to service the request.

***NO:** Program start requests are rejected if a prestart job is not immediately available when the program start request is received.

POOLID

Specifies the subsystem pool identifier under which the prestart jobs are run.

***SAME:** The value does not change.

pool-identifier: Specify the subsystem pool identifier under which prestart jobs run. Valid values range from 1 through 10.

CLS

Specifies the names of the classes under which the prestart jobs run and how many prestart jobs are allowed to run under each class. Jobs start by using the first class. After the number of jobs specified for the first class is reached, jobs are started under the second class. If the class does not exist when the entry is added, a library qualifier must be specified because the qualified class name is retained in the subsystem description.

Note: If the value specified for the MAXJOBS parameter is changed, the value for the number of jobs specified for this parameter might need to be changed. If *CALC or *MAXJOBS is specified, the system recalculates the value for the number of jobs to use the specified class.

Element 1: Class Name of First Class

***SAME:** The value does not change.

***SBSD:** The class having the same name as the subsystem description specified in the SBSD parameter is used for prestart jobs.

The name of the class can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

class-name: Specify the name of the class being used for prestart jobs. If no library qualifier is specified, the library list (*LIBL) of the job in which the CHGPJE command is run is used to find the class.

Element 2: Number of Jobs Using First Class

***SAME:** The value does not change.

***CALC:** The system calculates how many prestart jobs use this class. If one class is specified and *CALC is specified, all of the jobs use the specified class. If two classes are specified and *CALC is specified for both, the first class is the value of the MAXJOBS parameter divided by two, and the second class is the value of the MAXJOBS parameter minus the value calculated for the first class. If a specific number of jobs is specified for one class and *CALC is specified for the other class, the system calculates the difference between MAXJOBS and the specific number of jobs for the *CALC designation.

***MAXJOBS:** All prestart jobs use the specified class.

number-of-jobs: Specify the number of jobs that use this class. The value specified for both classes must total the value specified for the MAXJOBS parameter.

Element 3: Class Name of Second Class

***SAME:** The value does not change.

***NONE:** This value indicates that only one class is used.

***SBSD:** The class having the same name as the subsystem description, specified on the SBSD parameter, is used for prestart jobs.

The name of the class can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

class-name: Specify the name of the class being used for prestart jobs. If no library qualifier is specified, the library list (*LIBL) of the job where the CHGPJE command is run is used to find the class.

Element 4: Number of Jobs Using Second Class

***SAME:** The value does not change.

***CALC:** The system calculates how many prestart jobs use this class. If one class is specified and *CALC is specified, all of the jobs use the specified class. If two classes are specified and *CALC is specified for both, the first class is the value of the MAXJOBS parameter divided by two, and the second class is the value of the MAXJOBS parameter minus the value calculated for the first class. If a specific number of jobs is specified for one class and *CALC is specified for the other class, the system calculates the difference between MAXJOBS and the specific number of jobs for the *CALC designation.

***MAXJOBS:** All prestart jobs use the specified class.

number-of-jobs: Specify the number of jobs that use this class. The value specified for both classes must total the value specified for the MAXJOBS parameter.

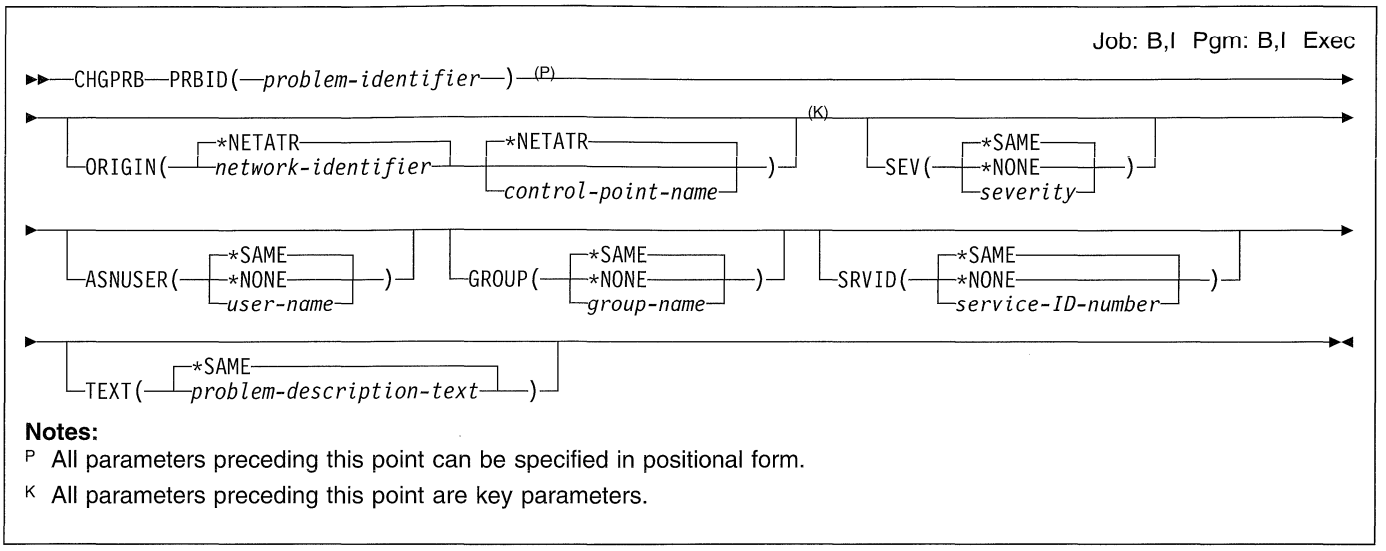
Example

```
CHGPJE  SBSD(QGPL/PJSBS)  PGM(QGPL/PGM1)  STRJOBS(*NO)
  THRESHOLD(1)  ADLJOBS(1)
```

This command changes the prestart job entry for the PGM1 program in the QGPL library in the PJSBS subsystem

description contained in the QGPL library. The prestart jobs associated with this entry are not started the next time the PJSBS subsystem description in the QGPL library is started. The STRPJ command is needed to start the prestart jobs. When more jobs need to be started, one additional job is started.

CHGPRB (Change Problem) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Problem command allows the user to change the values of selected fields within the problem log. The changeable fields include the service assigned number, problem severity, user name assigned to the problem log entry, and problem description.

Restriction: This command is shipped with public *EXCLUDE authority and the QPGMR, QSYSOPR, QSRV, and QSRVBAS user profiles have private authorities to use the command.

Required Parameter

PRBID

Specifies the problem identifier of the problem log entry. Problems with different system origins can have the same identifier. This parameter can be used with the ORIGIN parameter to select a single problem from a particular system origin.

Optional Parameters

ORIGIN

Specifies the node of the system from which the problem log entry originated. This parameter is used with the PRBID parameter to uniquely identify the problem.

Element 1: Network Identifier

***NETATR:** The LCLNETID value specified in the system network attributes is used.

network-identifier: Specify a network identifier.

Element 2: Control Point Name

***NETATR:** The LCLNETID value specified in the system network attributes is used.

control-point-name: Specify a control point name.

SEV

Specifies the severity level of the problem log entries being shown on the display. The severity level is assigned by the user when the problem is prepared for reporting. The four severity levels are:

1 High
2 Medium
3 Low
4 None

***SAME:** The value does not change.

***NONE:** No severity level is assigned to the problem.

severity: Specify the severity level. Valid values range from 1 through 4.

ASNUSER

Specifies the user name assigned to the problem log entry.

***SAME:** The value does not change.

***NONE:** No user name is assigned to the problem.

user-name: Specify the user name that is assigned to the problem log entry.

GROUP

Specifies the group in the filter to which the problem is assigned.

***SAME:** The group name does not change.

***NONE:** The problem is not assigned to a filter group.

group-name: Specify the group into which the problem entry is filtered.

SRVID

Specifies the service-assigned number for the problem log entry. This number is assigned when the problem is reported to IBM service support.

***SAME:** The value does not change.

***NONE:** No service-assigned number is assigned to the problem.

service-ID-number: Specify the service-assigned number for the problem log entry.

TEXT

Specifies text that briefly describes the problem. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

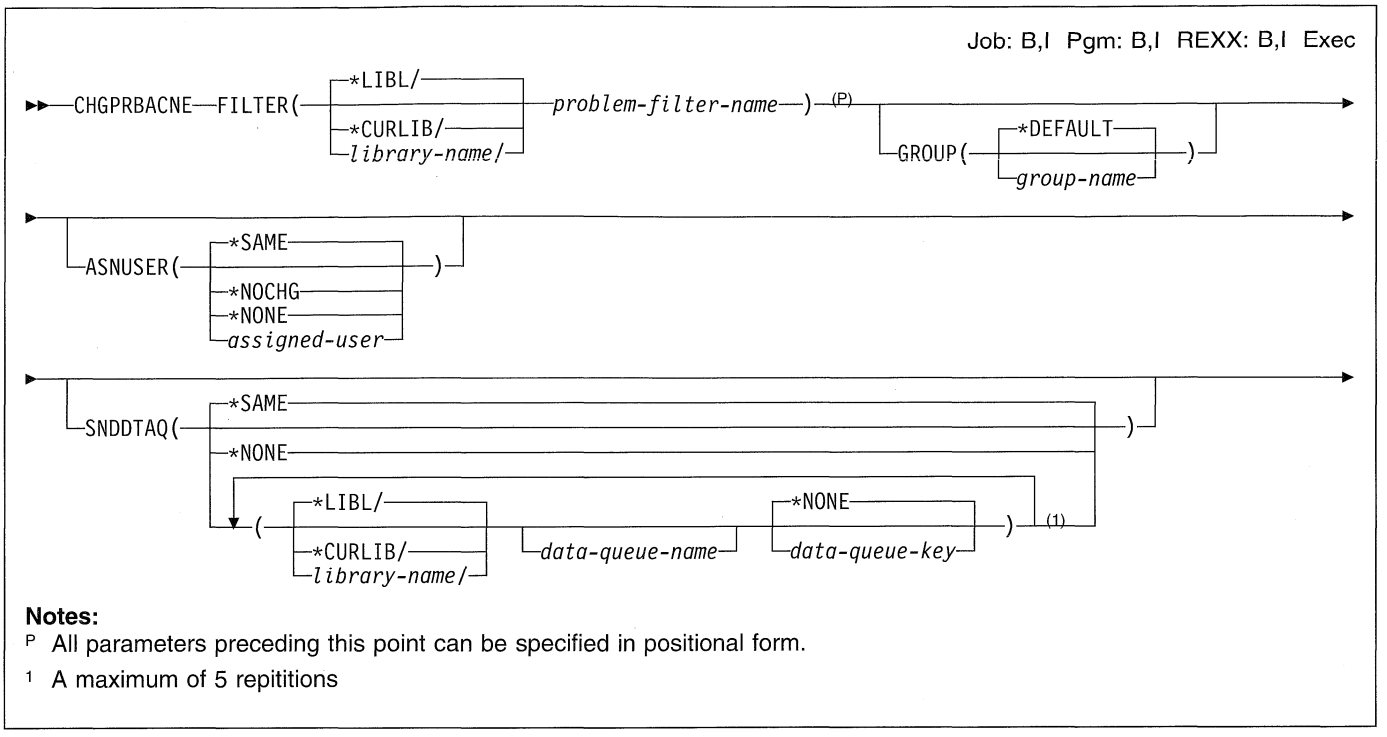
'problem-description-text': Specify the problem description.

Example

```
CHGPRB  PRBID(9213438081)  ORIGIN(AS400 SYSTEM02)
  SEV(4)  ASNUSER(JEFFREY)  GROUP(CHGPROB)
  SRVID(PMR01)  TEXT('New Problem Description')
```

This command changes the problem 9213438081 originating on SYSTEM02.AS400 to severity 4, changes the assigned user to JEFFREY, changes the group to CHGPROB, changes the service assigned number to PMR01, and adds a new description.

CHGPRBACNE (Change Problem Action Entry) Command



Notes:

P All parameters preceding this point can be specified in positional form.

1 A maximum of 5 repititions

Purpose

The Change Problem Action Entry (CHGPRBACNE) command allows you to change a problem action entry that was added using the Add Problem Action Entry (ADDPRBACNE) command.

Required Parameter

FILTER

Specifies the name of the filter.

The name of the filter can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

problem-filter-name: Specify the name of the filter.

Optional Parameters

GROUP

Specifies the group for which the actions are to be applied. The group name is assigned from selection criteria from a selection entry in the filter.

***DEFAULT:** The default specified when the filter was created is used.

group-name: Specify a group name.

ASNUSER

Specifies the user assigned to the problem log entry.

***SAME:** The value does not change.

***NOCHG:** No change occurs to the problem log entry.

***NONE:** No user is assigned to the problem log entry.

assigned-user: Specify a user name.

SNDDTAQ

Specifies the data queue for the problem notification record. Keyed data queues are supported.

***SAME:** The value does not change.

***NONE:** No data queue is used.

Element 1: Data Queue Name

The name of the data queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

data-queue-name: Specify the name of the data queue.

Element 2: Data Queue Key

***NONE:** No key is used on the data queue.

data-queue-key: Specify the data queue key.

An 80-byte record is enqueued on the data queue specified by the user. This record is received when the QRCVDTAQ program is called. The data queue does not have to be used solely for problems; alerts and problems can share the same data queue.

If a key is specified, it is used when enqueueing the record on the queue. If the data queue is non-keyed, the record is enqueued without a key.

Note: The time stamp used is the system standard time stamp. This time is already stored in the problem record.

The following table describes the record format.

Table 15. Record Format

Position	Type	Value	Description
1-10	CHAR	*PRBFTR	Problem filtering notification
11-11	CHAR	Function	Function performed 1–Problem created 2–Problem changed 3–Problem deleted
12-19	CHAR	Function TOD	TOD time stamp for function
20-29	CHAR	Group	Group problem was filtered into
30-39	CHAR	Problem ID	Problem ID number
40-59	CHAR	Origin System	System where problem originated
60-60	CHAR	Last Event	Last event committed into the history log (see note)
61-68	CHAR	Event TOD	TOD time stamp for Last Event
69-80	CHAR	Reserved	Reserved for future use

Note: Valid Last Event values are the following:

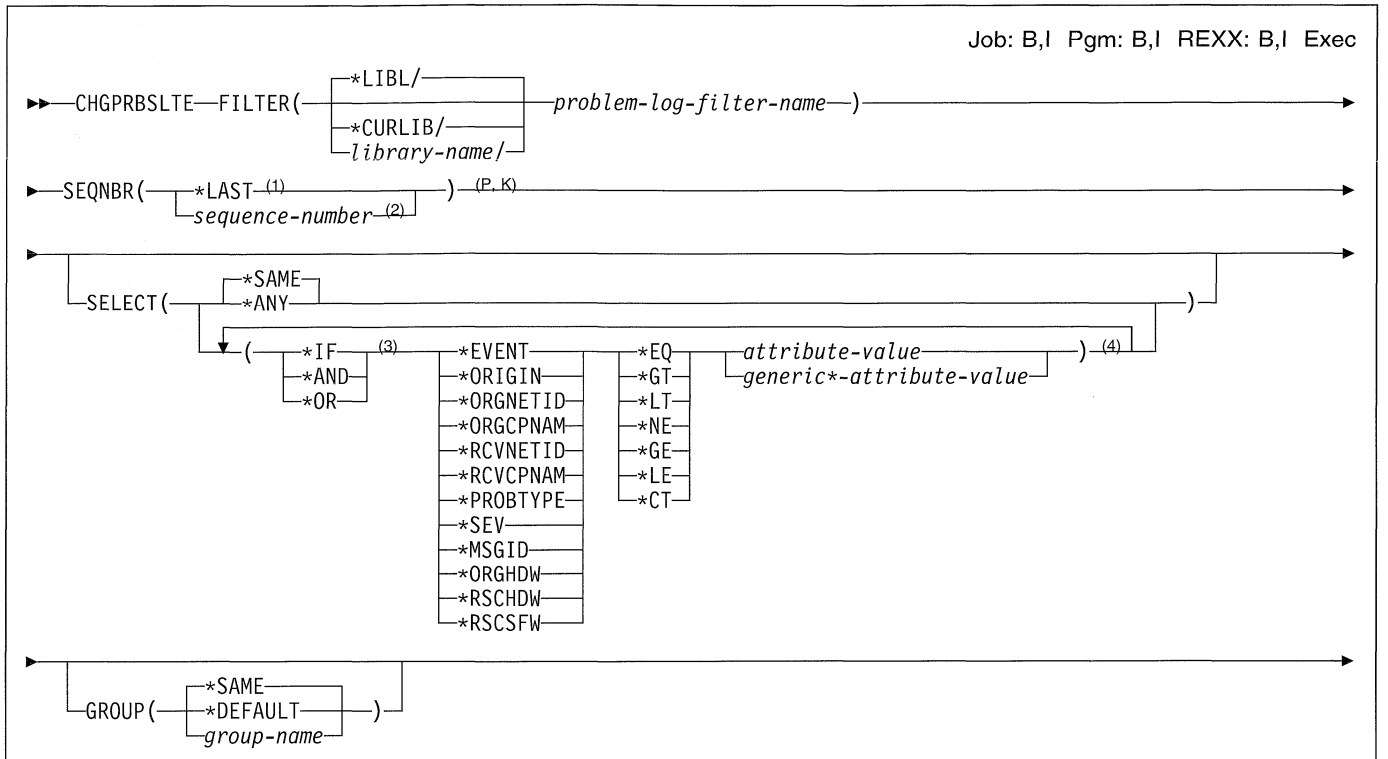
'01'X Problem entry opened
'02'X Request received
'03'X Opened by Alert
'10'X Problem analyzed
'11'X Verification test ran
'12'X Recovery procedure ran
'20'X Prepared to report
'21'X Service request sent
'22'X Problem answered
'23'X Response sent
'24'X Reported by voice
'25'X Fixes transmitted
'30'X Fix verified
'41'X Analyzed remotely
'42'X Remote verification ran
'43'X Remote recovery ran
'50'X Alert created
'51'X APAR created
'52'X APAR data saved
'54'X APAR data restored
'55'X APAR data deleted
'60'X Problem changed by Change Problem (CHGPRB) command
'61'X Problem deleted by Delete Problem (DLTPRB) command
'99'X Problem entry closed

Example

```
CHGPRBACNE   FILTER(MYLIB/MYFILTER)
  GROUP(IOWA)  ASNUSER(NIGHTOPR)
  SNDDTAQ(*SAME)
```

This command changes the actions for group IOWA. The assigned user is changed to NIGHTOPR.

CHGPRBSLTE (Change Problem Selection Entry) Command



Notes:

1 If *LAST is specified then SELECT must be *ANY or *SAME.

2 Range is from 1 through 9999

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

3 *IF must be specified as the first item only; either *AND or *OR must be specified when repeating selections.

4 A maximum of 10 repetitions

Purpose

The Change Problem Selection Entry (CHGPRBSLTE) command is used to change a problem selection entry that was added using the Add Problem Selection Entry (ADDPRBSLTE) command.

Required Parameters

FILTER

Specifies the name of the filter.

The name of the filter can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

problem-log-filter-name: Specify the name of the filter.

SEQNBR

Specifies the sequence number of the problem selection entry. Selection entries in a filter are numbered by sequence number. When a filter is applied, the selection entries with lower sequence numbers are tried first.

***LAST:** Change the default selection entry. This entry is automatically added when the filter is created.

sequence-number: Specify a number between 1 and 9999.

Optional Parameters

SELECT

Specifies that a problem log entry is selected or not selected based on whether information in the problem log entry satisfies a specified relationship.

You can specify a single value (*ANY) or all four elements that define a relationship. When you specify the

four elements, the attribute and attribute value are compared for the relationship specified by the relational operator.

***SAME:** The value does not change.

***ANY:** Any problem log entry is selected.

Element 1: Logical Operator

***IF:** The specified relationship must be satisfied for a problem log entry to be selected.

***AND:** The specified relationship must be satisfied in addition to the *IF relationship for a problem log entry to be selected.

***OR:** The specified relationship must be satisfied in addition to or instead of the *IF relationship for a problem log entry to be selected.

Element 2: Attribute

***EVENT:** The filter is applied when the problem log entry is created (a value of 1), changed (a value of 2), or deleted (a value of 3). If the entry has been created and is changed before being committed, use the value of 1.

***ORIGIN:** The problem log entry was locally generated (a value of L) or was received from another system (a value of R).

***ORGNETID:** The network identifier (ID) of the system in which the problem log entry originated is specified. This information is displayed using the Work with Problems (WRKPRB) command which shows the details for a specific problem. Specify the value in the following form:

```
'nnnnnnnnnn'
```

***ORGCPNAME:** The control point name of the system in which the problem log entry originated is specified. This information is displayed using the Work with Problems (WRKPRB) command which shows the details for a specific problem. Specify the value in the following form:

```
'ccccccccc'
```

***RCVNETID:** The network identifier of the remote system from which the problem log entry was received is specified. This information is displayed using the Work with Problems (WRKPRB) command which shows the details for a specific problem. Specify the value on the following form:

```
'nnnnnnnnnn'
```

***RCVCPNAM:** This attribute specifies the Remote System Control Point name in which the problem log entry received from. This information is displayed using the Work with Problems (WRKPRB) command and shows the details for a specific problem. The value specified for this attribute should be of the following form:

```
'ccccccccc'
```

***PROBTYPE:** The type of problem entry created. Possible problems are machine detected (a value of 1), user detected (a value of 2), PTF order (a value of 3), or application detected (a value of 4).

Note: User-Detected Remote Hardware problems are grouped with number 2 User-Detected problems.

***SEV:** The severity of the problem log entry created. Possible choices are high (a value of 1), medium (a value of 2), low (a value of 3), none (a value of 4), or not assigned (a value of 5).

Note: Problems do not have a severity level when locally created.

***MSGID:** The message ID found in the problem log entry. This is usually an AS/400 message ID from an AS/400 system.

***ORGHDW:** The origin hardware resource information in the problem log entry. This information is displayed using the Work with Problems (WRKPRB) command and shows the details for a specific problem. Specify the value in the following form:

```
'tttt mmm ss-ssssssss'
'tttt mmm ss-sssss'
'tttt mmm sssssss'
'tttt mmm sssss'
```

where tttt is the machine type, mmm is the model number and sssssssss is the serial number. Use this exact format to match a particular hardware resource exactly, or use a part of the hardware value with the Contains (*CT) relation to provide a partial match.

***RSCHDW:** The failing hardware resource information in the problem log entry. This information is displayed using the Work with Problems (WRKPRB) command and shows the details for a specific problem. Specify the value in the following form:

```
'tttt mmm ss-ssssssss'
'tttt mmm ss-sssss'
'tttt mmm sssssss'
'tttt mmm sssss'
```

where tttt is the machine type, mmm is the model number and sssssssss is the serial number. Use this exact format to match a particular hardware resource exactly, or use a part of the hardware value with the Contains (*CT) relation to provide a partial match.

***RSCSFW:** The failing software resource information in the problem log entry. This information is displayed using the Work with Problems (WRKPRB) command and shows the details for a specific problem. Specify the value in the following form:

```
'ppppppp vv rr mm'
```

where ppppppp is the licensed program ID, vv is the version number, rr is the release number, and mm is the modification level. Use this exact format to match a particular software resource exactly, or use a part of the software value with the Contains (*CT) relation to provide a partial match.

Element 3: Relational Operator

The value specified for Element 2 must have the following relationship to Element 4:

***EQ:** Equal to

***GT:** Greater than

***LT:** Less than

***NE:** Not equal to

***GE:** Greater than or equal to

***LE:** Less than or equal to

***CT:** Contains

Element 4: Attribute Value

attribute-value: Specify a value of up to 30 characters to compare with the contents of the attribute specified for Element 2. The value must be specified in character format and must be enclosed in apostrophes if it contains blanks or special characters. If a CL variable is specified for the value, it must be a character variable.

generic-attribute-value:* Specify the generic attribute value. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be changed only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

GROUP

Specifies the group to which a problem is assigned if it matches the criteria specified on the SELECT parameter.

***SAME:** The value does not change.

***DEFAULT:** The problem is assigned to the default group.

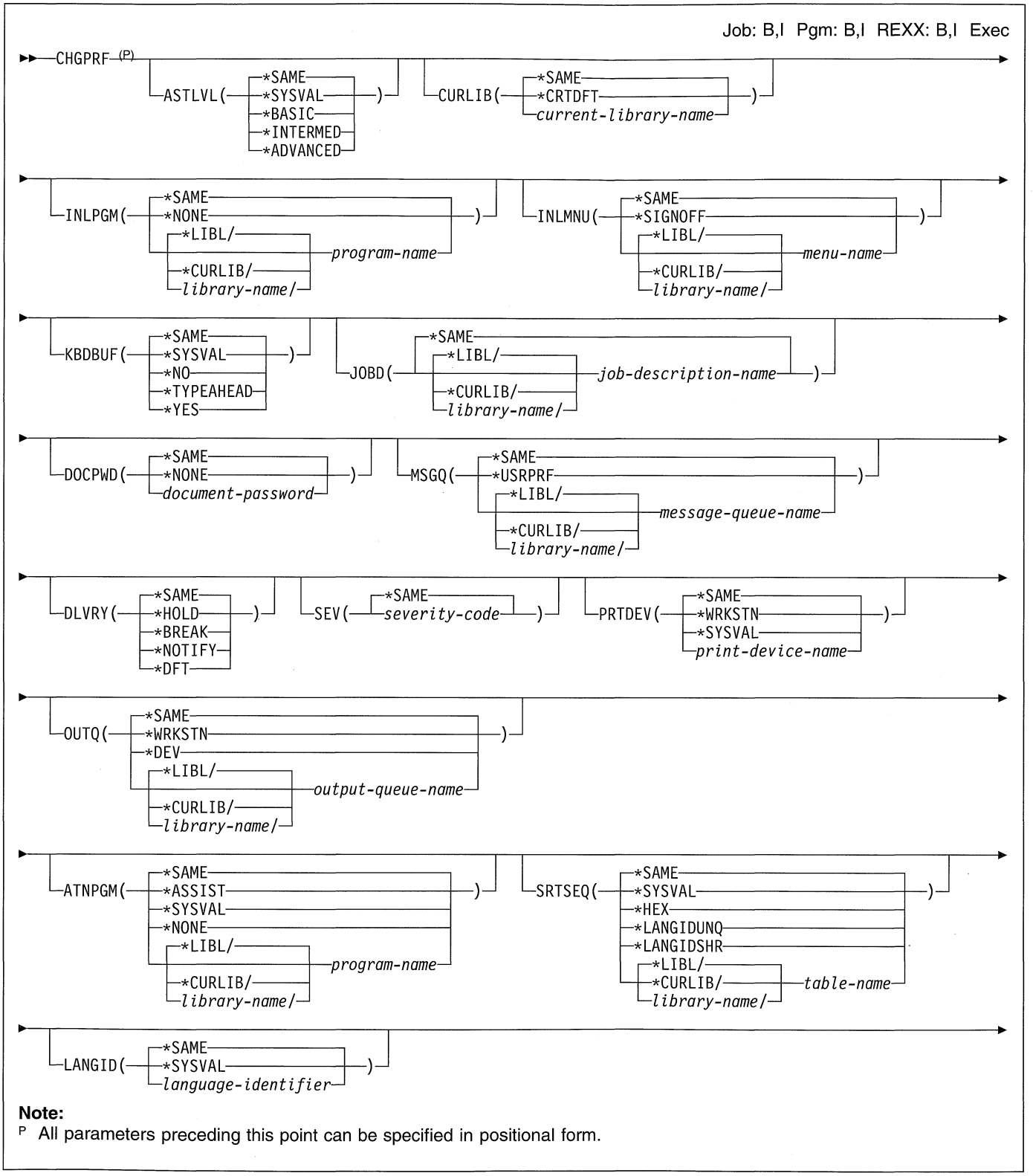
group-name: Specify a group name.

Example

```
CHGPRBSLTE  FILTER(MYLIB/MYFILT) SEQNBR(1250)
  SELECT((*IF *SEV *EQ 1) (*OR *SEV *EQ 2))
  GROUP(SEVHIGH)
```

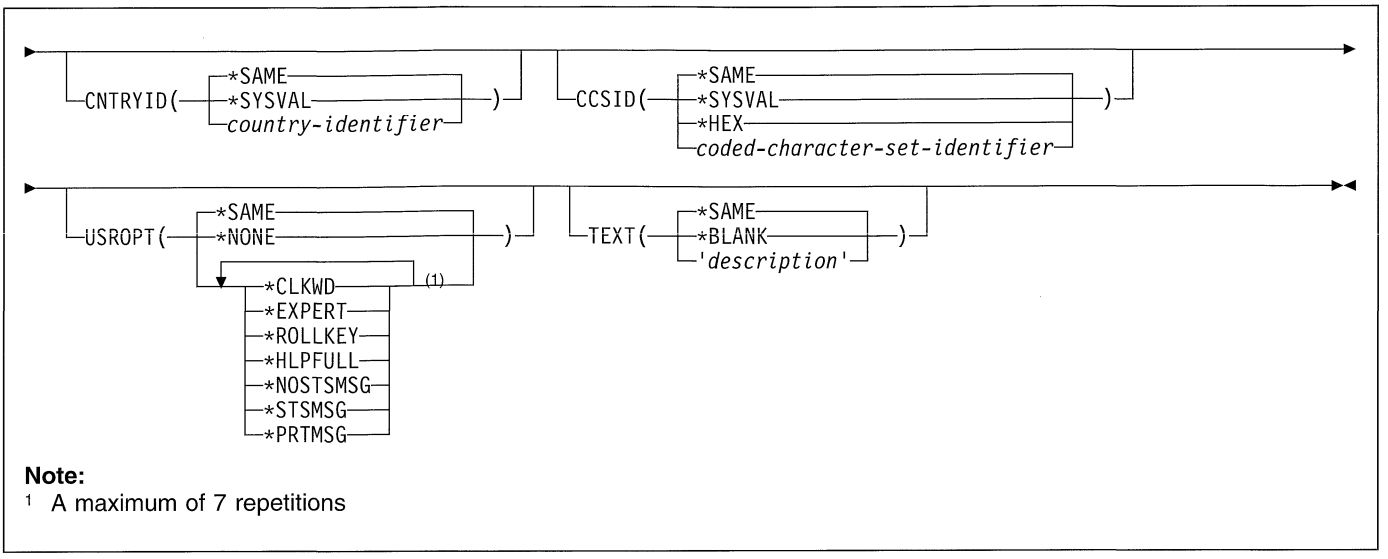
This command changes the problem selection entry with the sequence number 1250 in filter MYFILT located in library MYLIB. If the severity level equals 1 or 2, it is put in group SEVHIGH.

CHGPRF (Change Profile) Command



Note:

P All parameters preceding this point can be specified in positional form.



Note:
1 A maximum of 7 repetitions

Purpose

The Change Profile (CHGPRF) command allows a user to change some of the values currently specified in the user profile.

Restrictions:

1. Object management (*OBJMGT) authority and use (*USE) authority are required to change the user profile.
2. *USE authority is required to the current library, program, menu, job description, message queue, print device, output queue, and the ATTN key handling program.

Optional Parameters

ASTLVL

Specifies which user interface to use.

***SAME:** The value does not change.

***SYSVAL:** The assistance level defined for the system is used.

***BASIC:** The Operational Assistant user interface is used.

***INTERMED:** The system interface is used.

***ADVANCED:** The expert system interface is used. To allow for more list entries, the options keys and the function keys are not displayed. If a command does not have an advanced (*ADVANCED) level, the intermediate (*INTERMED) level is used.

CURLIB

Specifies the name of the library being used as the current library during the processing of this command.

Note: If *PARTIAL or *YES is specified for the LMTCPB parameter in the user profile, the user cannot change the current library.

***SAME:** The value does not change.

***CRTDFT:** The user has no current library. If objects are created into the current library using *CURLIB on a create command, the QGPL library is used as the default current library.

current-library-name: Specify the name of the library that, after the user signs on the system, becomes the current library.

INLPGM

Specifies, for an interactive job, the qualified name of the program that is called whenever a new routing step is started that has QCL or QCMD as the request processing program. If *PARTIAL or *YES is specified on the LMTCPB parameter in the user profile, the program value cannot be changed. No parameters can be passed to the program.

A system/36 environment procedure name can be specified as the initial program if the procedure is a member of the file QS36PRC (in the library list or specified library) and if either of the following conditions are true:

- *S36 is specified on the SPCENV parameter.
- *SYSVAL is specified on the SPCENV parameter and the system value, QSPCENV, is *S36.

***SAME:** The value does not change.

***NONE:** No program is called when the user signs on. If a menu name is specified in the INLMNU parameter, that menu is displayed.

The name of the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the program that is called when the user signs on.

INLMNU

Specifies the qualified name of the menu displayed when the user signs on. If *YES is specified on the LMTCPB parameter in the user profile, the user cannot change the menu.

A system/36 environment menu can be specified as the initial menu if either of the following conditions are true:

- *S36 is specified on the SPCENV parameter.
- *SYSVAL is specified on the SPCENV parameter and the system value, QSPCENV, is *S36.

***SAME:** The value does not change.

***SIGNOFF:** The system signs off the user after the program completes. This is intended for users authorized only to run the program.

The name of the menu can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

menu-name: Specify the name of the menu that is called when the user signs on.

KBDBUF

Specifies the keyboard buffering value to be used when a job is initialized for this user profile. The new value takes effect the next time the user signs on. If the type-ahead feature is active, the keystrokes can be buffered. If the attention key buffering option is active, the attention key is buffered like any other key. If the attention key is not active, the attention key is not buffered and is sent to the system even if the display station is input inhibited. The keyboard buffer value can also be set by a user application using the QWSSETWS program. More information is in the *System Programmer's Interface Reference* guide.

***SAME:** The value does not change.

***SYSVAL:** The system value, QKBDBUF, is used to determine the keyboard buffering value for this profile.

***NO:** The type-ahead feature and attention key buffering option are not active for this user profile.

***TYPEAHEAD:** The type-ahead feature is active for this user profile.

***YES:** The type-ahead feature and attention key buffering option are active for this user profile.

JOBD

Specifies the name of the job description used. If the job description does not exist when the entry is changed, a library qualifier must be specified because the qualified job description name is retained in the subsystem description. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The job description value does not change.

The name of the job description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

job-description-name: Specify the job description name used for the work station entries whose job description parameter values indicate the user JOBD(*USRPRF).

DOCPWD

Specifies the document password that allows Document Interchange Architecture (DIA) document distribution services users to protect personal distributions from being used by people working on their behalf.

***SAME:** The value does not change.

***NONE:** No document password is assigned to this user.

document-password: Specify a document password assigned to this user. The password must range from 1 through 8 alphanumeric characters (letters A through Z and numbers 0 through 9). The first character of the document password must be alphabetic; the remaining characters can be alphanumeric. Embedded blanks, leading blanks, and special characters are invalid.

MSGQ

Specifies the qualified name of the message queue to which messages are sent.

Note: The message queue is created, if it does not already exist, when the user signs on. If the message queue parameter is changed by using the change user profile command, then that message queue is also created.

***SAME:** The value does not change.

***USRPRF:** A message queue with the same name as that specified in the USRPRF parameter is used as the message queue for this user. This message queue is located in the QUSRSYS library.

The name of the message queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

message-queue-name: Specify the name of the message queue that is used by this user.

DLVRY

Specifies how messages sent to the message queue for this user are delivered.

***SAME:** The value does not change.

***HOLD:** The messages are held in the message queue until they are requested by the user or program.

***BREAK:** The job to which the message queue is assigned is interrupted when a message arrives at the message queue. If the job is an interactive job, the audible alarm is sounded (if the alarm feature is installed). The delivery mode cannot be changed to *BREAK if the message queue is also being used by another job.

***NOTIFY:** The job, to which the message queue is assigned, is notified when a message arrives at the message queue. For interactive jobs at a work station, the audible alarm is sounded (if the alarm feature is installed) and the Message Waiting light is turned on. The delivery mode cannot be changed to *NOTIFY if the message queue is also being used by another job.

***DFT:** The default reply to the inquiry message is sent. If no default reply is specified in the message description of the inquiry message, the system default reply, *N, is used.

SEV

Specifies the lowest severity code that a message can have and still be delivered to a user in break or notify mode. Messages arriving at the message queue whose severities are lower than the severity code specified on this parameter do not interrupt the job or turn on the audible alarm or the message-waiting light; they are held in the queue until they are requested by using the Display Message (DSPMSG) command. If *BREAK or *NOTIFY is specified on the DLVRY parameter, and is in effect when a message arrives at the queue, the message is delivered if the severity code associated with the message is equal to or greater than the value specified here. Otherwise, the message is held in the queue until it is requested.

***SAME:** The value does not change.

severity-code: Specify a severity code ranging from 00 through 99.

PRTDEV

Specifies the qualified name of the default printer device for this job. If OUTQ(*DEV) is specified, the file is placed on an output queue with the same name as the printer.

***SAME:** The value does not change.

***WRKSTN:** The printer assigned to the user's work station is used.

***SYSVAL:** The value specified in the system value QPRTDEV is used.

print-device-name: Specify the name of a printer used to print the output for this user.

OUTQ

Specifies the qualified name of the output queue.

***SAME:** The value does not change.

***WRKSTN:** The output queue assigned to the user's work station is used.

***DEV:** An output queue name with the same name as that specified in the PRTDEV parameter is used as the output queue for this user profile.

The name of the output queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

output-queue-name: Specify the output queue name that identifies the output queue associated with this user profile.

ATNPGM

Specifies the qualified name of the program that is the ATTN key handling program for this user. The ATTN key handling program is called when the ATTN key is pressed during an interactive job. The program is active only when the user routes to the system-supplied QCMD command processor. The ATTN key handling program is set on before the initial program (if any) is called and it is active for both program and menu. If the program changes the ATNPGM (by using the SETATNPGM command), the new program remains active only for the duration of the program. When control returns and QCMD calls the menu, the original ATTN key handling program becomes active again. If the SETATNPGM command is run from the menus or an application is called from the menus, the new ATTN key handling program that is specified overrides the original ATTN key handling program. If *YES or *PARTIAL is specified on the LMTCPB parameter in the user profile, the ATTN key handling program cannot be changed.

***SAME:** The value does not change.

***ASSIST:** QEZMAIN is used.

***SYSVAL:** The system value QATNPGM is used.

***NONE:** No ATTN key handling program is used by this user.

The name of the ATTN key handling program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the ATTN key handling program used by this user.

SRTSEQ

Specifies the sort sequence table to be used for string comparisons for this profile.

***SYSVAL:** The system value QSRTSEQ is used.

***HEX:** A sort sequence table is not used. The hexadecimal values of the characters are used to determine the sort sequence.

***LANGIDUNQ:** A unique-weight sort table is used.

***LANGIDSHR:** A shared-weight sort table is used.

The name of the sort sequence table can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

table-name: Specify the name of the sort sequence table to be used with this profile.

LANGID

Specifies the language identifier used for this user.

***SAME:** The language identifier does not change.

***SYSVAL:** The system value QLANGID is used.

language-identifier: Specify the language identifier. More information on valid language identifiers is in the *National Language Support Planning Guide*.

CNTRYID

Specifies the country identifier used for this user.

***SAME:** The country identifier does not change.

***SYSVAL:** The system value QCNTRYID is used.

country-identifier: Specify the country identifier from the country code table. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

CCSID

Specifies the coded character set identifier (CCSID) used for this user.

A CCSID is a 16-bit number identifying a specific set of encoding scheme identifiers, character set identifiers, code page identifiers, and additional coding-related information that uniquely identifies the coded graphic representation used.

Note: If the value for CCSID is changed, the change does not affect jobs that are currently running.

***SAME:** The CCSID does not change.

***SYSVAL:** The system value QCCSID is used.

***HEX:** The CCSID 65535 is used.

coded-character-set-identifier: Specify the CCSID. More information on valid CCSIDs is in the *National Language Support Planning Guide*.

USROPT

Specifies level of help information detail the user sees and the function of the Page Up and Page Down keys by default. The system shows several displays that are suitable for the inexperienced user. More experienced users must perform an extra action to see detailed information. When values are specified for this parameter, the system presents detailed information without further action by the experienced user.

***SAME:** The value does not change.

***NONE:** Detailed information is not shown.

***CLKWD:** Parameter keywords are shown instead of the possible parameter values when a control language (CL) command is displayed.

***EXPERT:** More detailed information is shown when the user is performing display and edit options to define or change the system (such as edit or display object authority).

***ROLLKEY:** The actions of the Page Up and Page Down keys are reversed.

***HLPFULL:** Help text is shown on a full display rather than in a window.

***NOSTSMSG:** Status messages are not displayed when sent to the user.

***STSMSG:** Status messages are displayed when sent to the user.

***PRTMSG:** A message is sent to this user's message queue when a spooled file for this user is printed or help by the printer writer.

TEXT

Specifies text that briefly describes the user profile named in the USRPRF parameter. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

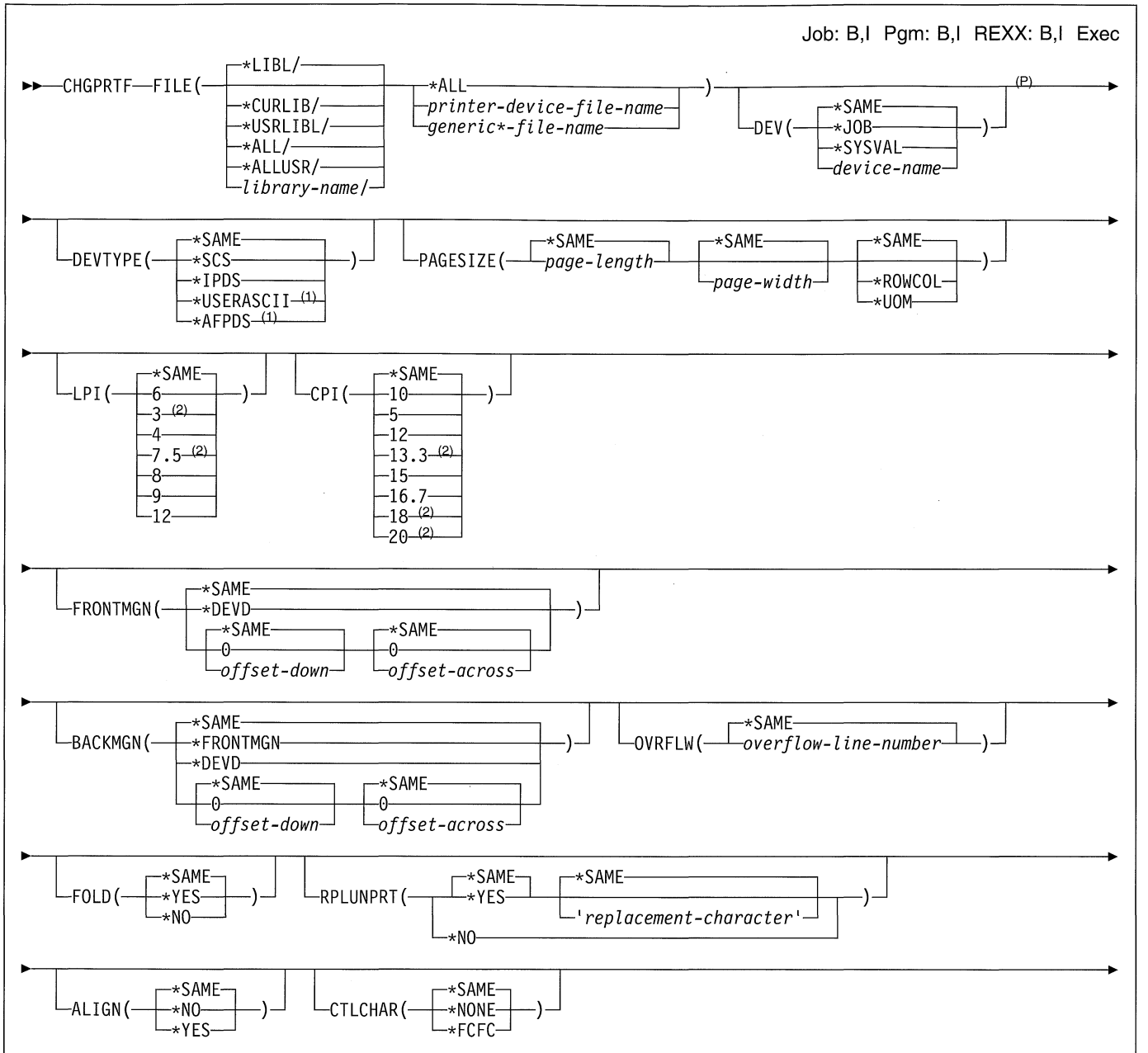
```
CHGPRF  INLPGM(ARLIB/DSPMENU)
```

In this example, JJADAMS wants to change his user profile.

This command makes the following changes to the user profile named JJADAMS:

- This command changes the program that is called following a successful sign-on to a program named DSPMENU, which is located in library ARLIB.
- All the other command parameters default to *SAME and are not changed.

CHGPRTF (Change Printer File) Command

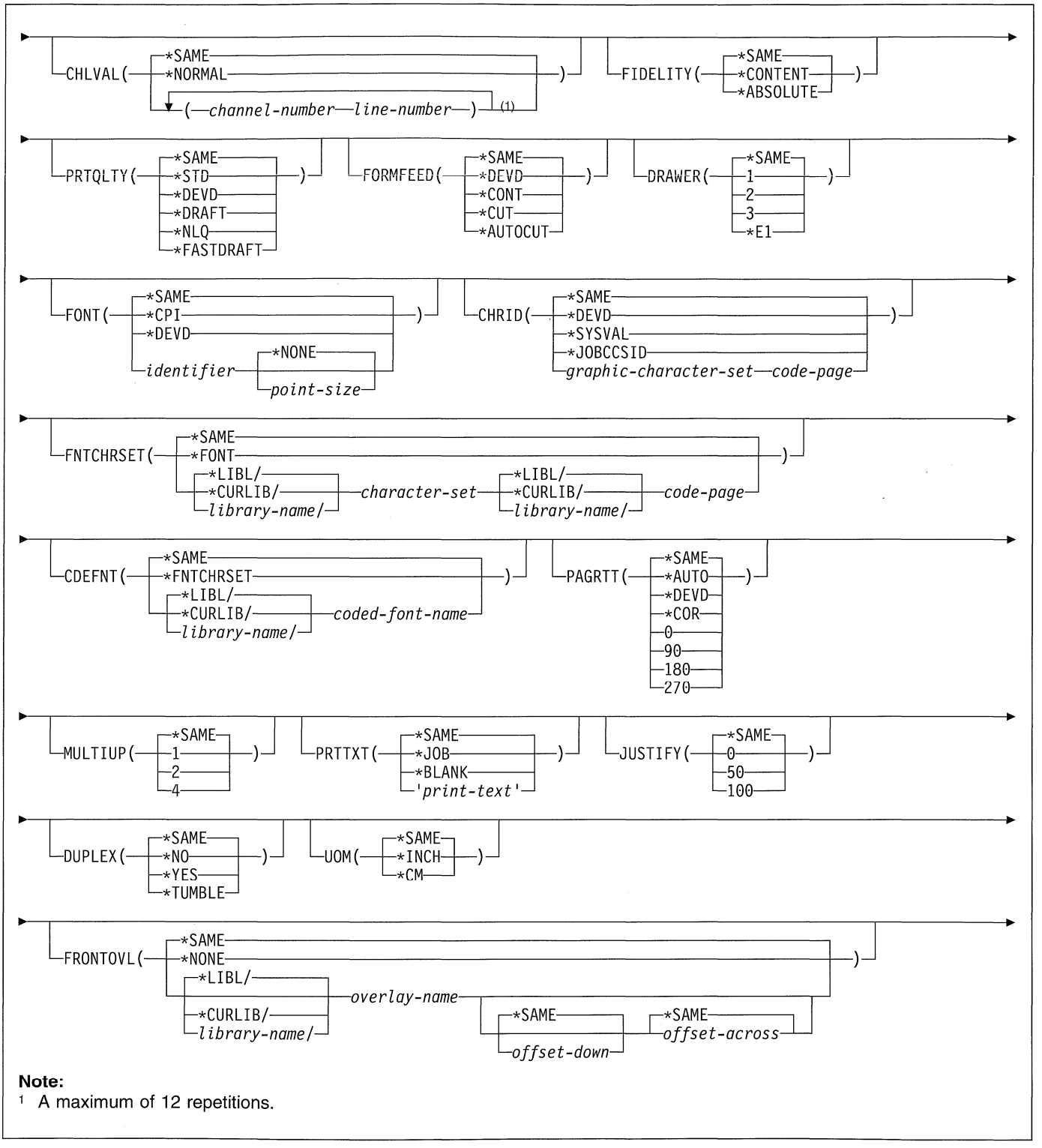


Notes:

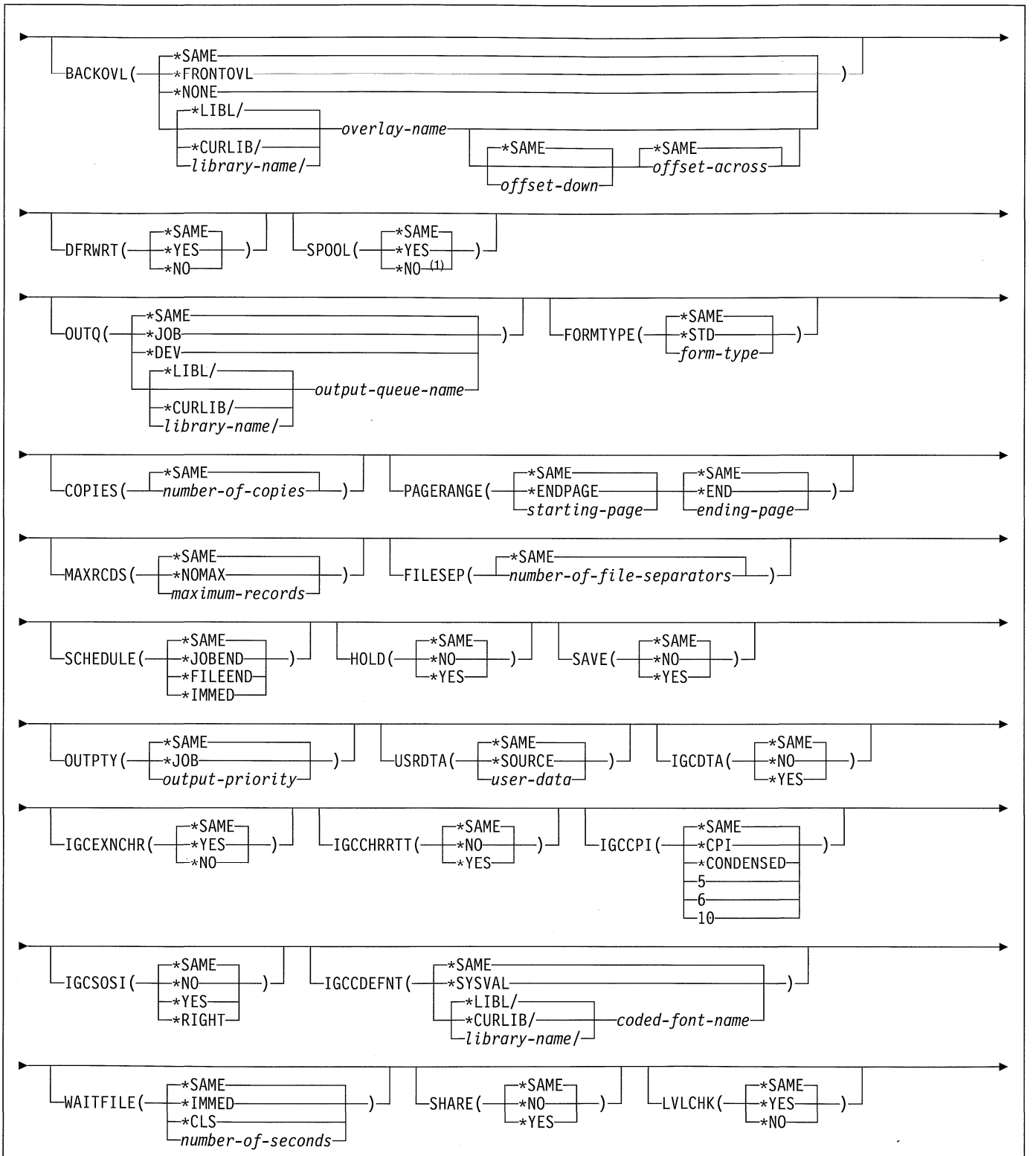
1 This value cannot be specified with SPOOL (*NO).

2 DBCS systems only.

P All parameters preceding this point can be specified in positional form.

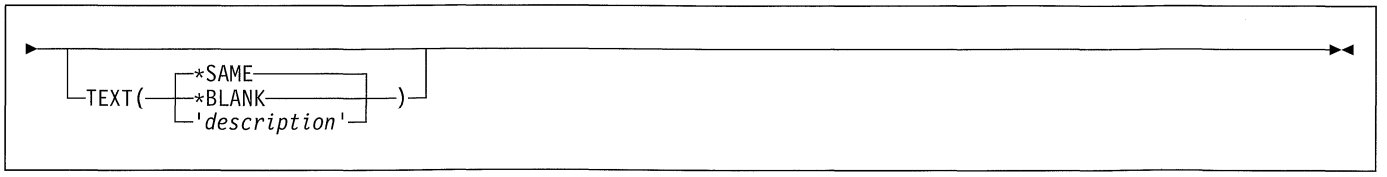


Note:
[1] A maximum of 12 repetitions.



Note:

[1] The value cannot be specified with DEVTYPE(*USERASCII) or DEVTYPE(*AFPDS).



Purpose

The Change Printer File (CHGPRTF) command changes, in the file description, one or more of the attributes of the specified printer device file. Only the information specified on this command can be changed. If the data description specifications (DDS) used to create the file are changed, the printer device file must be deleted and created again before the changes can be made in the file.

Required Parameter

FILE

Specifies the qualified name of the printer file whose description is being changed.

The name of the printer file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

***USRLIBL:** Only the libraries in the user portion of the job's library list are searched.

***ALL:** All libraries in the system portion of the job's library list, including QSYS, are searched.

***ALLUSR:** All user libraries are searched. All libraries with names that do not begin with the letter Q are searched except for the following:

```
#CGULIB    #DFULIB    #RPGLIB    #SEULIB
#COBLIB    #DSULIB    #SDALIB
```

Although the following Qxxx libraries are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are considered *user libraries*, and are also searched:

```
QDSNX      QPFRDATA   QUSER38
QGPL       QRCL       QUSRSYS
QGPL38     QS36F      QUSRVxRxMx
```

Note: A different library name, of the form QUSRVxRxMx, is added with each release. VxRxMx is the version, release, and modification level of the library.

library-name: Specify the name of the library to be searched.

***ALL:** All the printer files in the specified library are changed.

printer-device-file-name: Specify the name of the printer file being changed.

generic-file-name:* Specify the generic name of the printer device file. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be changed only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

Optional Parameters

DEV

Specifies the name of a printer device description. For nonspooled output, this identifies the printer device used with the printer file to produce the printed output. For spooled output, if OUTQ(*DEVD) is also specified, the default output queue for the specified printer is used for the spooled output data. More information is in the *Guide to Programming for Printing*.

***SAME:** The value does not change.

***JOB:** The printer device specified in the job description is used.

***SYSVAL:** The value specified in the system value QPRTDEV is used.

device-name: Specify the name of the printer associated with this display station. The printer and the display station must be attached to the same controller. When printing double-byte character set (DBCS) data, specify a DBCS printer (5553 or 5583).

DEVTYPE

Specifies the type of data stream created for a printer file.

***SAME:** The value does not change.

***SCS:** An SNA character stream (SCS) is created. This parameter must be specified when using the 3287, 3812 SCS, 3816 SCS, 4214, 4234 SCS, 4245, 5219, 5224, 5225, 5256, 5262, 6252, or 6262 work station printers.

- If *SCS is specified and the spooled printer file is directed to an IPDS printer, the SCS printer file is converted to emulate an IPDS printer file. More

information is in the *Guide to Programming for Printing*.

Double-Byte Character Set Consideration:

When using the 5553 and 5583 DBCS-capable printers, DEVTYPE(*SCS) must be specified.

***IPDS:** An intelligent printer data stream (IPDS) is created. This parameter can be specified when using a 3812 IPDS, 3816 IPDS, 3820, 3825, 3827, 3829, 3831, 3835, 3900, 4028, 4224, 4230, or the 4234 IPDS printer.

- If *IPDS is specified and the spooled printer file is directed to a printer other than an IPDS printer, the IPDS printer file is converted to an SCS printer file. More information is in the *Guide to Programming for Printing*.

***USERASCII:** An ASCII data stream is placed on a spooled output queue. The user is responsible for placing the entire hexadecimal data stream in the buffer, since the AS/400 system does not change or validate the values that are passed. This value cannot be specified with SPOOL(*NO).

***AFPDS:** An advanced function print data stream (AFPDS) is created. This parameter can be specified when using the 3812 IPDS, 3816 IPDS, 3820, 3825, 3827, 3829, 3831, 3835, 3900 4028, 4224, 4230, or the 4234 IPDS printer. The printer must be configured with AFP(*YES).

PAGESIZE

Specifies the length and width of the printer forms used by this printer file. The length is specified in lines per page or by the units specified for the UOM parameter. The width is specified in print positions (characters) per line or by the units specified for the UOM parameter.

The page size must be specified with reference to the way the data is printed on the page. For example, if using 8.5 inch wide by 11.0 inch long forms and printing at 6 lines per inch with a 10-pitch font, specify PAGESIZE(66 85) PAGRTT(0). However, to rotate the page, specify the page size for an 11.0 inch wide by 8.5 inch long page and enter PAGESIZE(51 110) PAGRTT(90).

Note: Specify PAGRTT(*AUTO) or PAGRTT(*DEVD) and PRTQLTY(*DRAFT) on this command to enable automatic reduction or rotation if the data does not fit on the paper.

Specify PAGRTT(*COR) on this command to enable automatic reduction whether or not the data fits on the paper.

Element 1: Page Length Value

***SAME:** The value does not change.

page-length: Specify the page length that is used by this printer file. Although a value ranging from 1 through 255 can be specified as the page length, the value specified must not exceed the actual length of the forms used.

More information about the page lengths that are valid for each printer type is in Appendix B, "Font, Character Identifier, and Other Values Supported for Different Printers."

Element 2: Page Width Value

***SAME:** The value does not change.

page-width: Specify the page width used by this printer file. The value specified must not exceed the actual width of the forms used.

More information about page width is in Appendix B, "Font, Character Identifier, and Other Values Supported for Different Printers."

Element 3: Method of Measure

***SAME:** The value does not change.

***ROWCOL:** Page length and page width are measured as numbers of rows and columns.

***UOM:** Page length and page width are measured in the units specified on the UOM parameter.

LPI

Specifies the line spacing setting on the printer, in lines per inch, used by this printer file.

The line spacing on the 5256 printer must be set manually. When the lines per inch (LPI) value on this parameter changes (from the value on the previous printer file), an inquiry message is sent to the message queue associated with the printer that requests a change to the LPI value.

The line spacing on the 4214, 4224, 4230, 4234, 4245, and 5262 Printers is set by a print command. These also allow setting the lines per inch spacing on the control panel of the printer. The lines per inch value must not be set at the printer. If the LPI value is overridden at the control panel, the system overrides the value set with the LPI value of the next printer file received.

More information about the lines per page and lines per inch that are valid for each printer type is in Appendix B, "Font, Character Identifier, and Other Values Supported for Different Printers."

***SAME:** The value does not change.

6: The line spacing on the printer is 6 lines per inch.

3: The line spacing on the printer is 3 lines per inch. This value is valid only for double-byte character set (DBCS) data.

4: The line spacing on the printer is 4 lines per inch.

7.5: The line spacing on the printer is 7.5 lines per inch. This value is valid only for double-byte character set (DBCS) printers.

8: The line spacing on the printer is 8 lines per inch.

Note: When printing double-byte character set (DBCS) data for a file specified with LPI(8), use double

spacing. Otherwise, the DBCS data does not print correctly. Alphanumeric data, however, prints correctly in single spacing when LPI(8) is specified.

9: The line spacing on the printer is 9 lines per inch.

12: The line spacing on the printer is 12 lines per inch.

CPI

Specifies the printer character density, in characters per inch (CPI), used by this printer file.

For the printers that support fonts, the value specified in the font special value implies the CPI. If FONT(*CPI) is specified, the font used is based on the CPI value. The following diagram describes the default font ID for each CPI value:

CPI	FONT ID DEFAULT
5	245
10	011
12	087
13.3	204
15	222
16.7	400
18	252
20	281

More information about the characters per page and characters per inch that are valid for each printer type is in Appendix B, "Font, Character Identifier, and Other Values Supported for Different Printers."

***SAME:** The value does not change.

10: Character density is 10 characters per inch.

5: Double-byte character density is 5 characters per inch.

12: Character density is 12 characters per inch.

13.3: Character density is 13.3 characters per inch. This value is valid only for double-byte character set (DBCS) printers.

15: Character density is 15 characters per inch.

16.7: Character density is 16.7 characters per inch.

18: Character density is 18 characters per inch. This value is valid only on double-byte character set (DBCS) printers.

20: Character density is 20 characters per inch. This value is valid only on double-byte character set (DBCS) printers.

FRONTMGN

Specifies the offset, down and across, of the origin from the edge on the front side of the paper. The offsets are in the units of measure specified on the UOM parameter. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57. This parameter can only be used for printer files with DEVTYPE(*AFPDS) specified.

***SAME:** The value does not change.

***DEVD:** The no-print border from the printer is used to place the text on the page when printing to a printer configured as AFP(*YES). A margin of 0 is used for IPDS printers without a no-print border, or which are configured as AFP(*NO).

Element 1: Offset Down

***SAME:** The value does not change.

0: No offset of the origin occurs.

offset-down: Specify the offset of the origin from the top of the page.

Element 2: Offset Across

***SAME:** The value does not change.

0: No offset of the origin occurs.

offset-across: Specify the offset of the origin from the left side of the page.

BACKMGN

Specifies the offset, down and across, of the origin from the edge on the back side of the paper. The offsets are in the units of measure specified on the UOM parameter. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57. This parameter can only be used for printer files with DEVTYPE(*AFPDS) specified.

***SAME:** The value does not change.

***FRONTMGN:** The offsets specified on the FRONTMGN parameter are used.

***DEVD:** The no-print border from the printer is used to place the text on the page when printing to a printer configured as AFP(*YES). A margin of 0 is used for IPDS printers without a no-print border, or which are configured as AFP(*NO).

Element 1: Offset Down

***SAME:** The value does not change.

0: No offset of the origin occurs.

offset-down: Specify the offset of the origin from the top of the page.

Element 2: Offset Across

***SAME:** The value does not change.

0: No offset of the origin occurs.

offset-across: Specify the offset of the origin from the left side of the page.

OVRFLW

Specifies the line number on the current page at which overflow to a new page begins. Generally, after the specified line is printed, the printer overflows to the next page before printing continues. Margins specified for the printer file are ignored when determining overflow. More information is in the *Guide to Programming for Printing.*

***SAME:** The value does not change.

overflow-line-number: Specify the line number on the current page at which overflow to a new page begins, whether or not printing has occurred on that line. The value specified must not be greater than the page length (PAGESIZE). Margins specified for the printer file are ignored when determining overflow.

FOLD

Specifies whether all positions in a record are printed when the record length exceeds the page width (specified by the PAGESIZE parameter). When folding is specified and a record exceeds the page width, any portion of the record that cannot be printed on the first line continues (is folded) on the next line or lines until the entire record has been printed.

The FOLD parameter is ignored under the following conditions:

- When DEVTYPE(*SCS) is not specified.
- When printing through OfficeVision/400.
- When in the S/36 execution environment.

Double-Byte Character Set Considerations:

The system ignores this parameter when printing double-byte character set (DBCS) files. The system assumes that DBCS records fit on a printed line. If the record exceeds the page width, the system continues printing the record on the next line.

***SAME:** The value does not change.

***YES:** Records are not folded; if a record is longer than the page width, only the part of the record that fits on one line is printed.

***NO:** Records whose length exceeds the page width are folded on the following lines.

RPLUNPRT

Specifies (1) whether unprintable characters are replaced and (2) which substitution character (if any) is used. An *unprintable* character is a character the printer is unable to print.

Double-Byte Character Set Considerations:

For double-byte character set (DBCS) data, an unprintable character is one that cannot be processed. When using DBCS-capable printers, consider the following:

- If IGCEXNCHR(*YES) is also specified, the system replaces unprintable extension characters with DBCS underline characters. There may be some cases in which the system is unable to replace an unprintable character with a DBCS underline character. In this case, the undefined character is printed.
- If IGCEXNCHR(*NO) is also specified, the device replaces all extension characters with the undefined character. Choosing a blank as the replacement character for alphanumeric characters might improve system performance.

More information is in the *Guide to Programming for Printing*.

Element 1: Replace Character?

***SAME:** The value does not change.

***YES:** Unprintable characters are replaced. The program is not notified when unprintable characters are detected. Note the DBCS considerations above.

Element 2: Replacement Character

***SAME:** The value does not change.

'replacement-character': Specify, if *YES is also specified on this parameter, the replacement character that is used each time an unprintable character is detected. Any printable EBCDIC character can be specified. Valid values range from 40 through 99 and A1 through FE.

Other Single Values

***NO:** Unprintable characters are not replaced. When an unprintable character is detected, a message is sent to the program.

ALIGN

Specifies whether the page must be aligned in the printer before printing is started. If ALIGN(*YES) and SPOOL(*NO) are specified, and forms alignment is required, the system sends a message to the message queue specified in the printer device description and waits for a reply to the message. When spool (*YES) is specified on the CRTPRTF command and ALIGN(*FILE) is specified on the STRPRTWTR command, then this parameter is used to determine whether an alignment message is sent by the system.

This parameter is ignored when cut sheets are used (spooled and direct output). Page alignment can be done only for text-only files. Page alignment cannot be done for print jobs containing graphics or bar codes.

***SAME:** The value does not change.

***NO:** No page alignment is required.

***YES:** The page is aligned before the output is printed.

CTLCHAR

Specifies whether the printer file supports input with print control characters. Any invalid control characters that are found are ignored, and single spacing is assumed.

***SAME:** The value does not change.

***NONE:** No print control characters are passed in the data being printed.

***FCFC:** The first character of every record contains an American National Standards Institute (ANSI) forms control character. If *FCFC is specified, the record length must include one extra position for the first-character forms-control code. This value is not valid for externally described printer files.

CHLVAL

Specifies a list of channel numbers with their assigned line numbers. Use this parameter only if CTLCHAR(*FCFC) has been specified.

Note: If one or more channel-number/line-number combinations are changed, all other combinations must be re-entered.

***SAME:** The value does not change.

***NORMAL:** The default values for skipping to channel identifiers are used. The default values are found in Table 16.

Table 16. ANSI First-Character Forms-Control Codes

Code	Action before Printing a Line
' '	Space one line (blank code)
0	Space two lines
-	Space three lines
+	Suppress space
1	Skip to line 1
2-11	Space one line
12	Skip to overflow line (OVRFLW parameter)

Element 1: Channel Number

channel-number: Specify an American National Standard channel number to be associated with a corresponding 'skip to' line number. Valid values for this parameter range from 1 through 12, corresponding to channels 1 through 12. The CHLVAL parameter associates the channel number with a page line number. For example, if you specify CHLVAL(2 20), channel identifier 2 is allocated with line number 20; therefore, if you place the forms-control 2 in the first position of a record, the printer skips to line 20 before printing the line.

Note: If the printer stops and the next record processed has a channel value forms-control number that is the same value as the line number the printer is on, the printer advances to that value (line number) on the next page. However, if the printer is positioned at the top of the page (line number one) and the channel value forms-control value is associated with line number one, the printer does not advance to a new a new page.

If no line number is specified for a channel identifier, and that channel identifier is encountered in the data, a default of 'space one line' before printing is used. Each channel number can be specified only once.

Element 2: Line Number

line-number: Specify the line number assigned for the channel number in the same list. Valid line numbers range from 1 through 255. If no line number is assigned to a channel number, and that channel number is encountered in the data, a default of 'space one line' before printing is used. Each line number can be specified only once.

FIDELITY

Specifies whether printing continues when print errors are found for printers configured with AFP(*YES).

***SAME:** The value does not change.

***CONTENT:** Printing continues when errors are found.

***ABSOLUTE:** Printing stops when errors are found.

PRTQLTY

Specifies, for the 3812 SCS, 3816 SCS, 4214, 4224, 4230, 4234, and 5219 printers, the quality of print produced.

For the 5219 Printer, different print qualities are produced by varying the speed at which the print ribbon advances. Quality mode (*STD or *NLQ) results in normal print ribbon advancement. In draft mode (*DRAFT), the ribbon advances at a rate of one-third the distance it advances in quality mode. The 5219 Printer has a conserve ribbon switch that overrides the value of *DRAFT specified by this parameter.

For the 3812 SCS and 3816 SCS Printers, the automatic hardware selection of computer output reduction printing selected through soft switches on the printers occurs only when *DRAFT is specified for PRTQLTY and PAGRTT is *DEVD. If PAGRTT(*COR) is specified, the PRTQLTY parameter does not affect the printed output.

For the 4214, 4224, 4230, and 4234 Printers, standard print quality is produced by varying the density of the dot matrix pattern used to create printable characters. Standard mode (*STD) is the normal mode. Quality mode (*NLQ) requires multiple passes by the printer to produce a line of data. Draft mode (*DRAFT) results in high-speed printing.

More information about the valid values for the 4214, 4224, 4230, 4234, and 5219 Printers is in Appendix B, "Font, Character Identifier, and Other Values Supported for Different Printers."

Notes:

1. For the 4214 Printer, standard print quality (*STD or *NLQ) is only supported for 10 and 12 characters per inch. If PRTQLTY(*STD or *NLQ) and 5, 15, or 16.7 characters per inch is specified, the data is printed in draft mode.
2. For the 4234 Printer, only a limited character set (62 characters) is supported when PRTQLTY(*DRAFT) is specified. A description of the character set supported with draft print quality is in the *4234 Printer Operator's Guide*.
3. For the 4224 and 4230 printers, the fonts supported are not available for all three print qualities. The OCR-A and OCR-B fonts are supported only with PRTQLTY(*NLQ). The Courier and Essay fonts are available only with PRTQLTY(*NLQ) and PRTQLTY(*STD). The Gothic font is available only

with PRTQLTY(*DRAFT) or PRTQLTY(*FASTDRAFT). If there is a mismatch between the print quality and the font selected, the font is changed to match the print quality.

4. Specify PAGRTT(*DEVD) and PRTQLTY(*DRAFT) on this command to enable automatic rotation if the data does not fit on the paper.

***SAME:** The value does not change.

***STD:** The output is printed with standard quality.

***DEVD:** The print quality is set on the printer by the user, if it is not set within the data stream.

***DRAFT:** The output is printed with draft quality.

***NLQ:** The output is printed with near letter quality.

***FASTDRAFT:** The output is printed at a higher speed and with lower quality than it would be if you specified *DRAFT. This value is only supported by the 4230 printer.

FORMFEED

Specifies, for the 4214, 5219, and 5553 printers (including ASCII printers that are configured as an SCS 4214 or SCS 5219 printer), and for IPDS printers, the mode in which forms are fed into the device.

***SAME:** The value does not change.

***DEVD:** The forms are fed into the printer in the manner specified in the device description.

***CONT:** Continuous forms are used by the printer. The tractor feed attachment must be on the device.

***CUT:** Single-cut sheets are used by the printer. Each sheet must be manually loaded. For cut sheets, the forms alignment message is not sent.

***AUTOCUT:** The sheet-feed attachment must be on the printer. Single-cut sheets are automatically fed into the printer. The forms alignment message is not sent for cut sheets.

DRAWER

Specifies the source drawer used when automatic cut sheet feed mode is used (specified by FORMFEED(*AUTOCUT)).

***SAME:** The value does not change.

1: The paper is fed from the first drawer on the sheet-feed paper handler.

2: The paper is fed from the second drawer on the sheet-feed paper handler.

3: The paper is fed from the third drawer on the sheet-feed paper handler.

***E1:** The envelopes are fed from the envelope drawer on the sheet-feed paper handler.

FONT

Specifies the font identifier and point size used with this printer device file. If a font identifier or point size is not specified, the system automatically sets them.

More information about the valid font identifiers, the display value, the characters per inch value implied with each font style, a description of each font style, and whether the font is supported on a particular printer is in Appendix B, "Font, Character Identifier, and Other Values Supported for Different Printers."

Note: Some fonts can be substituted by the printer. Consult the various printer reference guides for details.

***SAME:** The value does not change.

***CPI:** The identifier of the font with the specified pitch (characters per inch (CPI)) is used.

***DEVD:** The font identifier and point size specified in the device description are used.

Element 1: Font Identifier

identifier: Specify the numeric font identifier associated with this printer.

Element 2: Point Size

***NONE:** The point size is supplied by the system and is determined by the specified font identifier.

point-size: Specify a point size ranging from 0.1 through 999.9.

CHRID

Specifies the character identifier (graphic character set and code page) for the file. This parameter allows printing of text that is in different character identifier (graphic character set and code page) coding. The value specified on this parameter is used to instruct the printer device to interpret the hexadecimal byte string to print the same characters that were intended when the text was created. More information about the character identifier is in the *Guide to Programming for Printing*. A list of valid CHRID values and applicable printers is in the "CHRID Values and Applicable Printers (CHRID parameter)" table in Appendix B, "Font, Character Identifier, and Other Values Supported for Different Printers."

***SAME:** The value does not change.

***DEVD:** The default CHRID value that the device is designed to handle is used. The *DEVD value means character selection is normal because the file has the same character identifier as the device default.

***SYSVAL:** The system determines the graphic character set and code page values for the command parameters from the QCHRID system values.

***JOBCCSID:** The character identifier for the printer file is taken from the coded character set identifier (CCSID) of the job.

Note: This value is not allowed if the file was created on a system at an earlier release level than V2R3M0.

Element 1: Character Set

graphic-character-set: Specify the graphic character set values that match the attributes of the printer. Valid values range from 1 through 32767.

Element 2: Code Page

code-page: Specify the code page value that matches the attributes of the printer. Valid values range from 1 through 32767.

FNTCHRSET

Specifies a downloaded font consisting of a character set and code page. This parameter can only be used for printer files with DEVTYPE(*AFPDS) specified.

***SAME:** The value does not change.

***FONT:** The value specified on the FONT parameter is used.

Element 1: Font Character Set

The name of the font character set can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

character-set: Specify the font character set to use.

Element 2: Code Page Name

The name of the code page name can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

code-page: Specify the code page value used to create the command parameters. Valid values range from 1 through 999.

CDEFNT

Specifies the coded font that the system uses for single-byte character set (SBCS) printing. This parameter can only be used for printer files with DEVTYPE(*AFPDS) specified.

***SAME:** The value does not change.

***FNTCHRSET:** The font specified on the FNTCHRSET parameter is used.

The name of the coded font name can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

coded-font-name: Specify the DBCS-coded font name to use.

PAGRTT

Specifies the degree of text rotation for the 3812, 3816, 4028, 3820, 3825, 3827, 3829, 3831, 3835, and 3900 printers. This parameter allows the user to specify the degree of rotation of the text on the page with respect to the way the form is loaded into the printer. See the note under the PAGESIZE parameter for directions on specifying page size when rotating the page.

Specify *AUTO or *DEVD for this parameter and PRTQLTY(*DRAFT) on this command to enable automatic rotation if the data does not fit on the paper.

***SAME:** The value does not change.

***AUTO:** Indicates that automatic rotation of output is done to fit the printed data on the form. If rotation does not accomplish this, computer output reduction is performed automatically (regardless of the print quality being used). This parameter is valid only for printers supporting rotation.

***DEVD:** The operating system sends a device default rotation value to the printer. Page rotation is dependent upon your printer's specifications. See your printer or printer emulation documentation to determine how page rotation is affected.

***COR:** Computer output reduction is done. Computer output reduction allows printed output intended for a 13.2 inch wide by 11.0 inch long form to be printed on an 8.5 inch wide by 11.0 inch long form.

For computer output reduction printing, the following operations are done by the 3812, 3816, 4028, 3820, 3825, 3827, 3829, 3831, 3835, and 3900 printers:

- Automatic rotation to *COR is not done if the file contains graphics, bar codes, variable LPI, variable font, variable page rotations, or variable drawer.
- The text is rotated 90 degrees clockwise from the 0 degree rotation position (lower left corner of the first edge loaded into the printer).

  Note: For landscape paper on a 3835 printer, the rotation is counter-clockwise from the 0 degree rotation position (upper right corner of the first edge loaded into the printer).
- A top and left margin of 0.5 inches is added to the printed output.
- The 12-pitch fonts are changed to a 15-pitch font and 15-pitch fonts are changed to a 20-pitch font.

All other font widths are changed to a 13.3-pitch font, except for the 4028 printer where they are changed to a 15-pitch font.

- Vertical spacing (specified by the LPI parameter) is 70 percent of the normal spacing.
- The page size is set to 8.5 inches wide by 11 inches long.

0: No rotation occurs. Printing starts at and is parallel to the edge loaded into the printer.

90: Rotation of the text is done 90 degrees clockwise from the 0 degree writing position.

180: Rotation of the text is done 180 degrees clockwise from the 0 degree writing position.

270: Rotation of the text is done 270 degrees clockwise from the 0 degree writing position.

MULTIUP

Specifies, for spooled output only, the number of pages printed on a single physical page.

Note: Overlays are not reduced when more than one page is printed on a side.

***SAME:** The value does not change.

1: One page of output is printed on one physical sheet of paper.

2: Two pages of output are printed on 1 physical sheet of paper.

4: Four pages of output are printed on 1 physical sheet of paper.

PRTTXT

Specifies up to 30 characters of text to be printed at the bottom of each page of output. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***JOB:** The value for the current job is used.

***BLANK:** Text is not specified.

'print-text': Specify the character string printed at the bottom of each page. No more than 30 characters of text can be entered, enclosed in apostrophes.

JUSTIFY

Specifies the printing positions of the characters on a page so the right-hand margin of printing is regular. Justification is done to the record length on the printer file opened.

Note: The JUSTIFY parameter is supported only on the 3812 SCS, 3816 SCS, and 5219 Printers.

***SAME:** The value does not change.

0: No justification occurs.

50: Spaces are added to the blanks in the text so that the right margin is more closely aligned but not flush.

100: The text is expanded by spaces (added where the blanks already exist) until the right margin is flush.

DUPLEX

Specifies whether output is printed on one side or two sides of the paper.

***SAME:** The value does not change.

***NO:** The output is printed on one side of the paper.

***YES:** The output is printed on both sides of the paper with the top of each printed page at the same end of the paper.

***TUMBLE:** The output is printed on both sides of the paper with the top of one printed page at the opposite end of the sheet from the top of the other printed page. This is usually used for output that is bound at the top.

UOM

Specifies the unit of measure that is used.

***SAME:** The value does not change.

***INCH:** An inch is used as the unit of measure.

***CM:** A centimeter is used as the unit of measure.

FRONTOVL

Specifies the qualified name of the object that contains both the overlay that is printed on the FRONT side of the page and the offset, down and across, from the point of origin used when the overlay is printed.

***SAME:** The value does not change.

***NONE:** No overlay is used.

Element 1: Overlay Name

The name of the overlay can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

overlay-name: Specify the name of the overlay.

Element 2: Offset Down

***SAME:** The value does not change.

offset-down: Specify the offset down from the point of origin at which to begin printing the overlay. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57.

Element 3: Offset Across

***SAME:** The value does not change.

offset-across: Specify the offset across from the point of origin at which to begin printing the overlay. If UOM(*CM) is specified, valid values range from 0

through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57.

BACKOVL

Specifies the object name and library name containing both the overlay that is printed on the BACK side of the page and the offset, down and across, from the point of origin used when the overlay is printed.

***SAME:** The value does not change.

***FRONTOVL:** The values that are specified on the FRONTOVL parameter are used.

***NONE:** No overlay is used.

Element 1: Overlay Name

The name of the overlay can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

overlay-name: Specify the name of the overlay.

Element 2: Offset Down

***SAME:** The value does not change.

offset-down: Specify the offset down from the point of origin at which to begin printing the overlay. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57.

Element 3: Offset Across

***SAME:** The value does not change.

offset-across: Specify the offset across from the point of origin at which to begin printing the overlay. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57.

DFRWRT

Specifies how much output is held in the system buffer before being sent to the printer.

***SAME:** The output queue associated with the job is used.

***YES:** The system controls the amount of output that is held in the buffer before being sent to the printer.

If SPOOL(*YES) is specified along with SCHEDULE(*IMMED), output is held in the buffer until a page of output is available or until the system buffer is full.

***NO:** If SPOOL(*NO) is specified, output is not held in the buffer. Output is sent to the printer immediately after the program performs a write operation.

If the spooled output schedule is not immediate, specifying DFRWRT(*NO) has no effect.

SPOOL

Specifies whether the output data for the printer file is spooled. If SPOOL(*NO) is specified, the following parameters in this command are ignored: OUTQ, COPIES, MAXRCDS, FILESEP, SCHEDULE, HOLD, SAVE, OUTPTY, and USRDTA.

***SAME:** The value does not change.

***YES:** The data is spooled for processing by a diskette writer or a print writer.

***NO:** The data is not spooled; it is sent directly to the device and printed as the output becomes available.

OUTQ

Specifies, for spooled output only, the qualified name of the output queue.

***SAME:** The value does not change.

***JOB:** The output queue associated with the job is used.

***DEV:** The output queue associated with the printer specified on the DEV parameter is used.

The name of the output queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

output-queue-name: Specify the name of the output queue to which the output data is spooled.

FORMTYPE

Specifies the type of form on which the output is printed. The identifiers used to indicate the type of forms are user-defined and can be a maximum of 10 characters in length.

***SAME:** The value does not change.

***STD:** The standard form type is used.

form-type: Specify the identifier of the form type used with this device file for printed output from jobs. Up to 10 alphanumeric characters can be specified. When the device file is opened, the system sends a message identifying the form type to the system operator, and requests that the identified forms be in the printer.

COPIES

Specifies, for spooled files, the number of copies being printed.

***SAME:** The value does not change.

number-of-copies: Specify a value, ranging from 1 through 255, that indicates the number of identical printouts produced when this printer file is used.

PAGERANGE

Specifies the page range to print for each copy of the file to be printed.

Element 1: Starting Page to Print

***SAME:** The value does not change.

***ENDPAGE:** Only the ending page is printed.

starting-page: Specify the page on which to start printing.

Element 2: Ending Page to Print

***SAME:** The value does not change.

***END:** The last page in the file is printed.

ending-page: Only the ending page is printed.

MAXRCDS

Specifies, for spooled output only, the maximum number of records that can be in the spooled file for jobs using this printer file. If this maximum is reached, an inquiry message is sent to the program message queue.

***SAME:** The value does not change.

***NOMAX:** The system maximum is used.

maximum-records: Specify a value, ranging from 1 through 999999, that specifies the maximum number of records allowed in the spooled file.

FILESEP

Specifies, for spooled output only, the number of separator pages placed at the start of each printed file, including those between multiple copies of the same output. Each separator page has the following items printed on it: file name, file number, job name, user name, and the job number.

***SAME:** The value does not change.

number-of-file-separators: Specify the number of separator pages used at the start of each printed output file produced by this device file. Valid values range from 0 through 9. If 0 is specified, no separator pages are printed for the file. In this case, the printed output for each file (or copy of a file) starts at the top of a new page.

SCHEDULE

Specifies, for spooled output only, when the spooled file is available to a writer.

***SAME:** The value does not change.

***JOBEND:** The spooled file is made available to the writer only after the entire job is completed.

***FILEEND:** The spooled file is made available to the writer as soon as the file is closed in the program.

***IMMED:** The spooled file is made available to the writer as soon as the file is opened in the program.

HOLD

Specifies, for spooled output only, whether the spooled file is held. The spooled file can be released by using the Release Spooled File (RLSSPLF) command.

***SAME:** The value does not change.

***NO:** The spooled printer file is not held by the output queue. The spooled output is available to a writer based on the SCHEDULE parameter value.

***YES:** The spooled file is held until released by the Release Spool File (RLSSPLF) command.

SAVE

Specifies, for spooled output only, whether the spooled file is saved (left on the output queue) after the output has been produced.

***SAME:** The value does not change.

***NO:** The spooled file data is not saved on the output queue after it has been produced.

***YES:** The spooled file data is saved on the output queue until the file is deleted. After the file is produced, the number of copies (see COPIES parameter) is set to 1, and its status is changed from WTR to SAV. Refer to the Release Spooled File (RLSSPLF) command for information on how to produce the spooled file again.

OUTPTY

Specifies the output priority for spooled output files that are produced by this job. The highest priority is 1 and the lowest priority is 9. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***JOB:** The output priority associated with the job that created the spooled file is used.

output-priority: Specify the output priority. Valid values range from 1 (high priority) through 9 (low priority).

USRDTA

Specifies, for spooled output only, the user-specified data that identifies the file.

***SAME:** The value does not change.

***SOURCE:** If the spooled file was created by an application program, the name of the program is used. Otherwise, blanks are used.

user-data: Specify up to 10 characters of text.

IGCDTA

Specifies, for program-described original files, whether the file processes double-byte character set (DBCS) data. For externally described printer files, this parameter specifies DBCS attributes of the file.

For program-described files:

***SAME:** The value does not change.

***NO:** The file does not process DBCS data.

***YES:** The file processes DBCS data.

For externally described files:

***SAME:** The value does not change.

***NO:** The only DBCS attributes of the file are those defined in the data description specifications (DDS).

***YES:** The file processes DBCS data.

IGCEXNCHR

Specifies whether the system processes double-byte character set (DBCS) extension characters.

***SAME:** The value does not change.

***YES:** The system processes DBCS extension characters.

***NO:** The system does not process DBCS extension characters; it prints extension characters as the undefined character.

IGCCHRRTT

Specifies, for the 5553 and 5583 Printers only, whether the printer rotates double-byte characters 90 degrees counterclockwise when printing. The system prints rotated double-byte characters so they appear in a vertical reading sequence. Alphanumeric characters are not rotated.

***SAME:** The value does not change.

***NO:** The system does not rotate double-byte characters when printing.

***YES:** The system rotates double-byte characters 90 degrees counterclockwise when printing. The printer rotates each character individually.

IGCCPI

Specifies the printer character density of double-byte character set (DBCS) characters, in characters per inch (CPI).

Note: This parameter does not specify the printer character density of alphanumeric characters. Alphanumeric characters are printed with the value specified on the CPI parameter.

***SAME:** The value does not change.

***CPI:** DBCS character density is based on the values specified for the CPI parameter. The system prints one double-byte character for every two alphanumeric characters.

- For CPI(10), DBCS characters print at 5 characters per inch.
- For CPI(12), DBCS characters print at 6 characters per inch.
- For CPI(13.3), DBCS characters print at 6.7 characters per inch (same as IGCCPI(*CONDENSED)).
- For CPI(15), DBCS characters print at 7.5 characters per inch.
- For CPI(18), DBCS characters print at 9 characters per inch.
- For CPI(20), DBCS characters print at 10 characters per inch.

***CONDENSED:** Condensed printing is used in which the system prints 20 DBCS characters every 3 inches. This value is valid only for the 5553 or 5583 Printers.

5: Double-byte character density is 5 characters per inch.

6: DBCS character density is 6 characters per inch. This value is valid for the 5553 and 5583 Printers only.

10: DBCS character density is 10 characters per inch. This value is valid for the 5553 or 5583 Printers only.

IGCSOSI

Specifies, for bracketed DBCS character strings only, how the system prints shift control characters.

***SAME:** The value does not change.

***NO:** The system does not print shift control characters. These characters do not occupy a position in printed output.

***YES:** The system prints shift control characters as blanks.

***RIGHT:** The system prints two blanks when printing shift-in characters but does not print shift-out characters.

IGCCDEFNT

Specifies the coded font that the system uses for DBCS printing.

***SAME:** The value does not change.

***SYSVAL:** The DBCS-coded font specified in the system value QIGCCDEFNT is used.

The name of the coded font name can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

coded-font-name: Specify the coded font name to use.

WAITFILE

Specifies the number of seconds that the program waits for the file resources and session resources to be allocated when the file is opened, or for the device or session resources to be allocated when an acquire operation is performed to the file. If those resources are not allocated within the specified wait time, an error message is sent to the program. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

Note: An immediate allocation of the device by the device resource is required when an acquire operation is performed to the file.

***SAME:** The value does not change.

***IMMED:** The program does not wait; when the file is opened, an immediate allocation of the file resources is required.

***CLS:** The job default wait time is used as the wait time for the file resources being allocated.

number-of-seconds: Specify the number of seconds that the program waits for the file resources to be allocated to the printer file when the file is opened, or the wait time for the device allocated when an acquire operation is performed to the file. Valid values range from 1 through 32767 seconds.

SHARE

Specifies whether the open data path (ODP) for the printer file is shared with other programs in the routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

More information on shared database files is in the *Database Guide*.

***SAME:** The value does not change.

***NO:** The ODP created by the program with this attribute is not shared with other programs in the routing step. Every time a program opens the file with this attribute, a new ODP to the file is created and activated.

***YES:** The ODP created with this attribute is shared with each program in the routing step that also specifies SHARE(*YES) when it opens the file.

Note: When SHARE(*YES) is specified and control is passed to a program, a read operation in that program retrieves the next input record. A write operation produces the next output record.

LVLCHK

Specifies whether the record format level identifiers in the program are checked against those in the device file when the file is opened. If so, the record format identifiers in the program must match those in the device file. Because the same record format name can exist in more than one file, each record format is given an internal system identifier when it is created.

***SAME:** The value does not change.

***YES:** The level identifiers of the record formats are checked when the file is opened. If the level identifiers do not match, an error message is sent to the program that requested the open, and the file is not opened.

***NO:** The level identifiers are not checked when the file is opened.

TEXT

Specifies text that briefly describes the printer file. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Examples

Example 1: Changing Two Parameters

```
CHGPRTF   FILE(ACCREC/PRTRPT)  LPI(6)  ALIGN(*YES)
```

This command changes two parameters in printer file PRTRPT stored in library ACCREC. The system operator must align the pages in the printer before the system starts printing the file. The file is printed in 6 lines per inch on the pages.

Example 2: Changing All IBM-Supplied Printer Files

```
CHGPRTF  FILE(QSYS/Q*)  PAGESIZE(88 132)  LPI(8)
  OVRFLW(80)
```

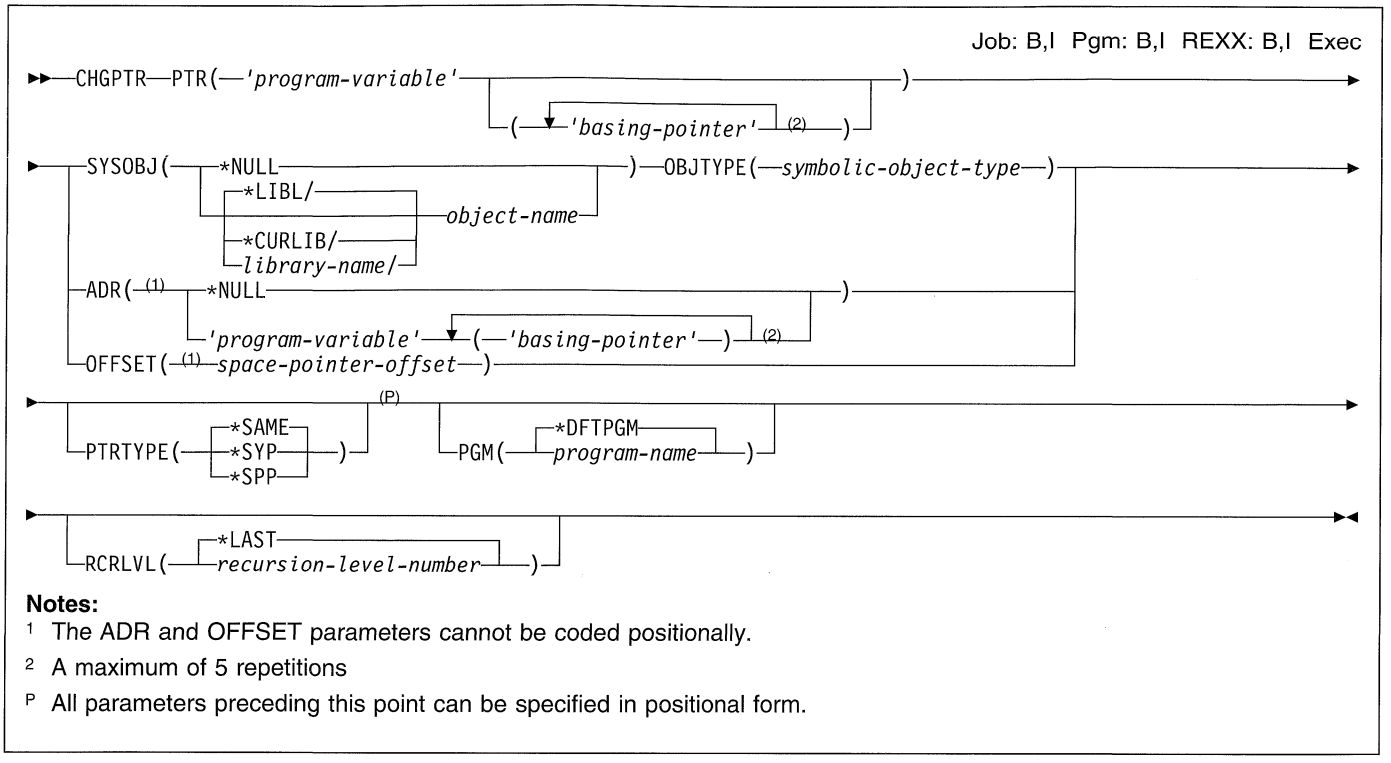
This command changes all IBM-supplied printer files (that is, all printer files in library QSYS whose names start with a Q) to use 88 lines of 132 characters (8 lines per inch), but to skip to the next page after 80 lines.

Example 3: Processing DBCS Data

```
CHGPRTF   FILE(IGCLIB/IGCPRT)  FORMFEED(*AUTOCUT)
  IGCDTA(*YES)  IGCCHRRTT(*YES)
```

This command changes printer file IGCPRT stored in library IGCLIB, so that it processes double-byte character set data. The system rotates double-byte characters before printing, and cut sheets are fed automatically when printing.

CHGPTR (Change Pointer) Command



Notes:

1 The ADR and OFFSET parameters cannot be coded positionally.

2 A maximum of 5 repetitions

P All parameters preceding this point can be specified in positional form.

Purpose

The Change Pointer (CHGPTR) command changes the value of a pointer variable in a program. The value of the program pointer specified can be changed to point to a new system object (SYSOBJ), to a new space pointer address (ADR), or to a new offset within a space object (OFFSET). This command is not normally used in high-level language (HLL) programs.

Restrictions:

1. This command is shipped with public *EXCLUDE authority, and the QSRV user profile has private authority to use the command.
2. This command is valid only for changing program variables that are used as pointers and is valid only in debug mode. To start debug mode, refer to the STRDBG (Start Debug) command.
3. This command cannot be used if the user is servicing another job, and that job is on a job queue, or is being held, suspended, or ended.
4. This command cannot be used to change variables in a bound program.
5. This command cannot be used to change variables that are write-protected or within the system domain, unless the user has *SERVICE special authority.

Required Parameters

PTR

Specifies the name of the pointer (program variable) whose value is being changed, allowing the pointer to point to a different address.

If the pointer variable is an HLL pointer (which was declared in the source for a high-level language program), a space pointer or null value can be assigned to the pointer, or the offset of the pointer can be changed, but the variable cannot be set with a system pointer value. If the user specifies a pointer that is not an HLL pointer, or if the user specifies an object-definition-table-vector (ODV) number for a pointer which is an HLL pointer, this restriction does not apply. More information on testing and debugging at the machine interface level is in the *CL Programmer's Guide*.

Element 1: Program Variables

'program-variable': Specify the name of the pointer variable whose value is being changed. The name must be enclosed in apostrophes if it contains special characters.

If the pointer (program variable) is an array, the subscripts representing the element in the array to be changed must be specified. Up to 132 characters may be specified for this pointer (program variable) entry. This includes any qualifiers, subscripts, embedded blanks, parentheses, and commas. It does not include the enclosing apostrophes when special characters are not used. An integer, machine-interface object-definition-table-vector (MI ODV) number, or a numeric

variable name can be specified for a subscript. For more information on the program-variable value, see the Appendix C, "Parameter Values Used for Testing and Debugging."

Some examples are:

```
PTR('A(2,INDEX2)')
PTR(PTR3)
```

Element 2: Basing Pointers

'basing-pointer': Specify the basing pointers for the pointer being changed. In some languages, the pointer (program variable) can be based on a pointer variable. This set of values allows the user to specify the basing pointers for the pointer being changed. Each basing pointer name must be enclosed in apostrophes if contains special characters.

If the basing pointer is an array, the subscripts representing an element in the array must be specified. Up to 132 characters can be specified for a basing pointer name. This includes any qualification, embedded blanks, parentheses, and commas. It does not include the enclosing apostrophes when special characters are used. An integer, MI ODV number, or a numeric variable name can be specified for a subscript. For more information on the basing pointer value, refer to Appendix C, "Parameter Values Used for Testing and Debugging."

Some examples are:

```
PTR('PTR1(B,5)' 'PTR2(C)')
PTR(PTR1 (BASEPTRA BASEPRT2))
```

SYSOBJ

Specifies that the pointer is set to either a system pointer or to a space pointer that addresses a particular system object, or to a null pointer value. This parameter cannot be specified when an HLL pointer is specified for the PTR parameter.

***NULL:** The system pointer is set to a null; that is, it no longer points to any system object nor does it have a pointer type. The OBJTYPE parameter cannot be specified if *NULL is specified here.

The name of the object can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

object-name: Specify the name of the object to which the system pointer is set. The pointer variable is set to one of the following:

- A system pointer value (if PTRTYPE(*SYP) is specified, or if PTRTYPE(*SAME) is used and the pointer variable already contains a system pointer)
- A space pointer value (if PTRTYPE(*SPP) is specified or if PTRTYPE(*SAME) is used and the pointer variable already contains a space pointer)

The user who specifies this value must have at least *USE authority for the specified object.

OBJTYPE

Specifies the object type of the system object specified in the SYSOBJ parameter to which the pointer named in the PTR parameter is set. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

ADR

Specifies the name of the program variable (if any) to which the specified space pointer is to point (that is, the program variable's address).

***NULL:** The space pointer is set to a null; it no longer points to the address of any space object nor does it have a pointer type.

Element 1: Program Variables

'program-variable': Specify the name of the program variable to which the space pointer is set. The name must be enclosed in apostrophes if it contains special characters.

If an array is specified without any subscripts, the pointer is set to the address of the first element in the array. Up to 132 characters may be specified for this program variable entry. This includes any qualifiers, subscripts, embedded blanks, parentheses, and commas. It does not include the enclosing apostrophes when special characters are used. An integer, MI ODV number, or numeric variable name can be specified for a subscript. For more information on the program-variable value, refer to Appendix C, "Parameter Values Used for Testing and Debugging."

Some examples are:

```
ADR('VAR1(INX1,3)')
ADR(DSPARRAY)
```

Element 2: Basing Pointers

'basing-pointer': In some languages, the program variable may be based on a pointer variable. This set of values allows the user to specify the basing pointers for the variable that is addressed. Each basing pointer name must be enclosed in apostrophes if it contains special characters.

If the basing pointer is an array, the subscripts representing an element in the array must be specified. Up to 132 characters can be specified for a basing pointer name. This includes any qualifiers, embedded blanks, parentheses, and commas. It does not include the enclosing apostrophes when special characters are used. An integer, MI ODV number, or a numeric vari-

able name can be specified for a subscript. For more information on the basing pointer value, refer to Appendix C, "Parameter Values Used for Testing and Debugging."

Some examples are:

```
ADR('VAR1(VAR2,5)' 'PGMPTR(5)')
ADR(VAR5 (BASEPTRA BASEPRT2))
```

OFFSET

Specifies the value to which the offset portion of the specified space pointer is set. Specify the number of bytes from the start of the space object to which the space pointer is set.

Optional Parameters

PTRTYPE

Specifies the type of pointer to which the pointer named in the PTR parameter is set.

An HLL pointer cannot be changed to a system pointer value. HLL pointers should be changed by using the CHGHLLPTR command.

***SAME:** The value does not change.

***SYP:** The pointer type is a system pointer.

***SPP:** The pointer type is a space pointer.

PGM

Specifies the name of the program that contains the pointer whose value is to change.

***DFTPGM:** The program previously specified as the default program contains the pointer whose value is to change.

program-name: Specify the name of the program that contains the pointer whose value is to change. The same name must already have been specified in the Start Debug (STRDBG) command or Add Program (ADDPGM) command.

RCRLVL

Specifies which recursion level of the program contains the pointer whose value is being changed. Changes made to static variables automatically affect all recursion levels. Recursion level 1 is the first (or earliest) call of the program, recursion level 2 is the second call of the program, and so on to the last (most recent) recursion level in the stack. For example, if program A calls program B, and then program B calls program A, a new recursion level of program A is formed. If the first recursion level of program A contains the pointer being changed, RCRLVL(1) must be specified. Some high-level languages also allow recursive procedures. The appropriate high-level language manual has more information on these programs.

***LAST:** The last, most recent, call of the specified program contains the pointer being changed.

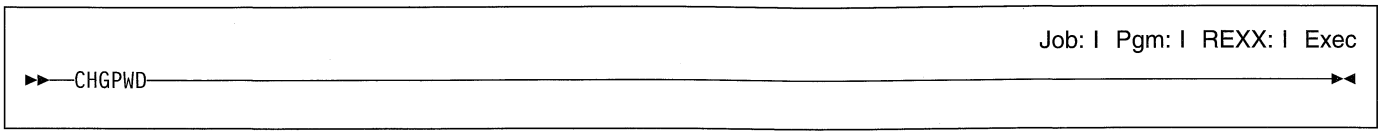
recursion-level-number: Specify the number of the recursion level of the program that contains the pointer being changed.

Example

```
CHGPTR   PTR(DATAFILPTR)  SYSOBJ(QGPL/MYFILE)
  OBJTYPE(*FILE)
```

This command changes the value of the pointer DATAFILPTR that is used in the default program in the debug mode. The pointer value is changed to point to the file called MYFILE, which is stored in the QGPL library.

CHGPWD (Change Password) Command



Purpose

The Change Password (CHGPWD) command shows the Change Password display allowing the user to change the password. The password is the security key that allows the user to sign on the system.

The new password that is entered from the change password display is checked against the password validation rules. The password validation rules are defined by AS/400 system values. A description of the password validation rules is in *Security Reference*.

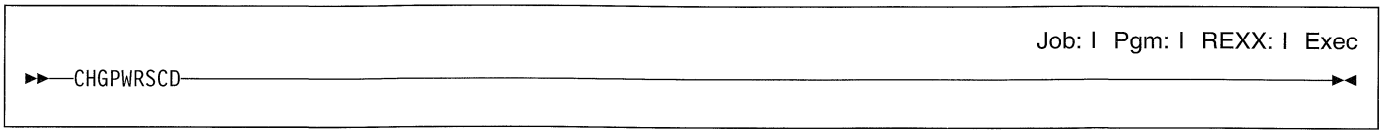
There are no parameters for this command.

Example

```
CHGPWD
```

This command shows the user the Change Password display.

CHGPWRSCD (Change Power On/Off Schedule) Command



Purpose

The Change Power On/Off Schedule (CHGPWRSCD) command is used to display and change the system's power on/off schedule from the Change Power On/Off Schedule display. On the display, the user can change power on or power off default values for the days of the week or change the values for a particular day. The user can also change or set the time the system sends a message that warns users of an impending power off. The changes the user makes in the power on/off schedule are effective immediately.

Restrictions: The user of this command must have *ALLOBJ and *SECADM special authority and must have authority to the Power Down System (PWRDWNSYS) command.

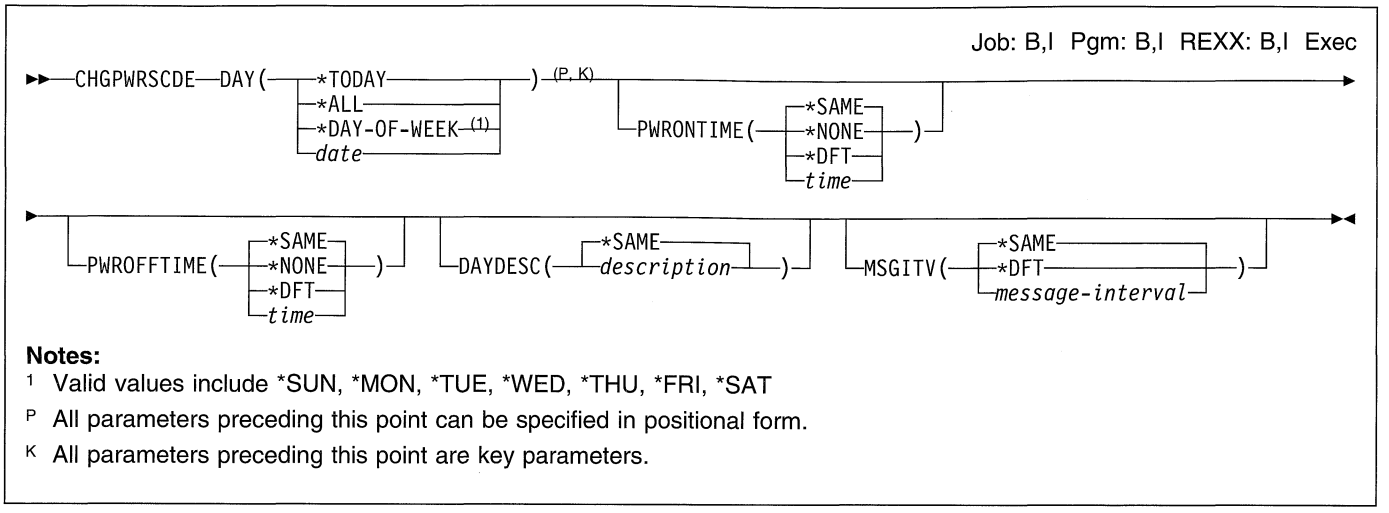
There are no parameters for this command.

Example

```
CHGPWRSCD
```

This command displays the Change Power On/Off Schedule display.

CHGPWRSCDE (Change Power On/Off Schedule Entry) Command



Notes:

1 Valid values include *SUN, *MON, *TUE, *WED, *THU, *FRI, *SAT

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Power On/Off Schedule Entry (CHGPWRSCDE) command is used to change the system's power on/off schedule. The user can change power-on or power-off default values for the days of the week or change the values for a particular day. The user can also change or set the time the system sends a message that warns users of an impending power off. The changes the user makes in the power on/off schedule are effective immediately.

Restrictions: The user of this command must have *ALLOBJ and *SECADM special authority and authority to the Power Down System (PWRDWNSYS) command.

Required Parameter

DAY

Specifies the day or days for which the user wants to change the power on/off schedule.

***TODAY:** The schedule for the current date is changed.

***ALL:** The default values for all days of the week are changed.

***DAY-OF-WEEK:** The default values for the specified day of the week are changed.

date: Specify the date to be changed. The date must be specified in the format defined by the job.

Optional Parameters

PWRONTIME

Specifies the power-on time.

***SAME:** The value does not change.

***NONE:** No power-on time or default power-on time is set.

***DFT:** The power-on time for the date being changed is set to the default value for the day of the week on which the date occurs. This value is allowed only if *TODAY or a specific date is specified on the DAY parameter.

time: Specify the power-on time. The time is specified in 24-hour format with or without a time separator as follows:

- With a time separator, specify a string of 5 or 8 digits where the time separator separates the hours, minutes, and seconds. If this command is entered from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command fails.
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds. Valid values for **hh** range from 00 through 23. Valid values for **mm** and **ss** range from 00 through 59.

PWROFFTIME

Specifies the power-off time.

***SAME:** The value does not change.

***NONE:** No power-off time or default power-off time is set.

***DFT:** The power-off time for the date being changed is set to the default value for the day of the week on which the date occurs. This value is allowed only if *TODAY or a date is specified on the DAY parameter.

time: Specify the power-off time. See the PWRONTIME parameter for a description of time formats.

DAYDESC

Specifies the description for the day. The user can enter an explanation of the day's power on/off schedule. This parameter is allowed only if *TODAY or a date is specified on the DAY parameter.

***SAME:** The value does not change.

'description': Specify up to 38 characters of text for an explanation of the day's power on/off schedule.

MSGITV

Specifies the number of minutes before the scheduled power off that a message is sent to all work stations warning users of the impending power off. This parameter is allowed only if DAY(*ALL) is specified.

***SAME:** The value does not change.

***DFT:** A message is sent 30 minutes prior to the power off.

message-interval: Specify the message interval. Valid values range from 0 through 60 minutes.

Examples

Example 1: Changing the Schedule For An Entire Week

```
CHGPWRSCDE   DAY(*ALL)  PWRONTIME(0800)
  PWROFFTIME(1800)
CHGPWRSCDE   DAY(*SAT)  PWRONTIME(*NONE)
  PWROFFTIME(*NONE)
CHGPWRSCDE   DAY(*SUN)  PWRONTIME(*NONE)
  PWROFFTIME(*NONE)
```

These commands set the power on and power off values for an entire week.

Example 2: Changing the Power-Off Time

```
CHGPWRSCDE   DAY('01/22/90')  PWROFFTIME(2000)
CHGPWRSCDE   DAY('01/22/90')  PWROFFTIME(2000)
 PWRONTIME(*SAME)
```

Either of these commands is used to set the power-off time to 8 p.m. on January 22, 1990.

Example 3: Changing the Power-On Time

```
CHGPWRSCDE   DAY(012590)  PWRONTIME(060000)
  PWROFFTIME(*NONE)
```

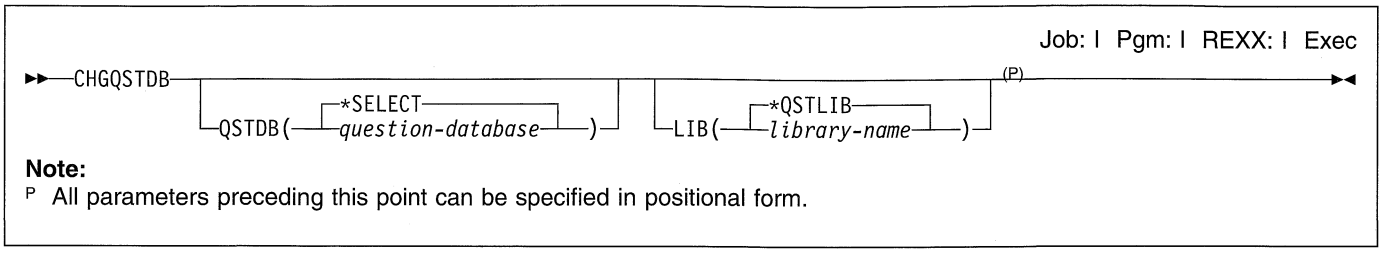
This command sets the power-on time to 6 a.m. for January 25, 1990 and sets no power-off time.

Example 4: Changing Back to the Defaults

```
CHGPWRSCDE   DAY(012590)  PWROFFTIME(*DFT)
  PWRONTIME(*DFT)
```

This command sets the power on and off times for January 25, 1990 back to the defaults for that day of the week.

CHGQSTDB (Change Question-and-Answer Database) Command



Purpose

The Change Question-and-Answer Database (CHGQSTDB) command allows the user to change the characteristics, topics, or search words of a question-and-answer (Q & A) database. When the user enters this command, a menu appears where the user can select the part of the Q & A database to change.

More information is in the *Q & A Database Coordinator's Guide*.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. A user must have authority to the command and be a Q & A coordinator for any Q & A database referred to by the command.
3. This command can only be used interactively.

Optional Parameters

QSTDB

Specifies the Q & A database to change.

***SELECT:** The user is asked to specify a Q & A database. If only one Q & A database exists on the system, it is the default.

question-database: Specify the name of the Q & A database to change.

LIB

Specifies the name of the library that contains the Q & A database.

***QSTLIB:** The library containing the specified Q & A database is searched. If *SELECT is specified on the QSTDB parameter, any Q & A database in any library for which the user is authorized can be selected.

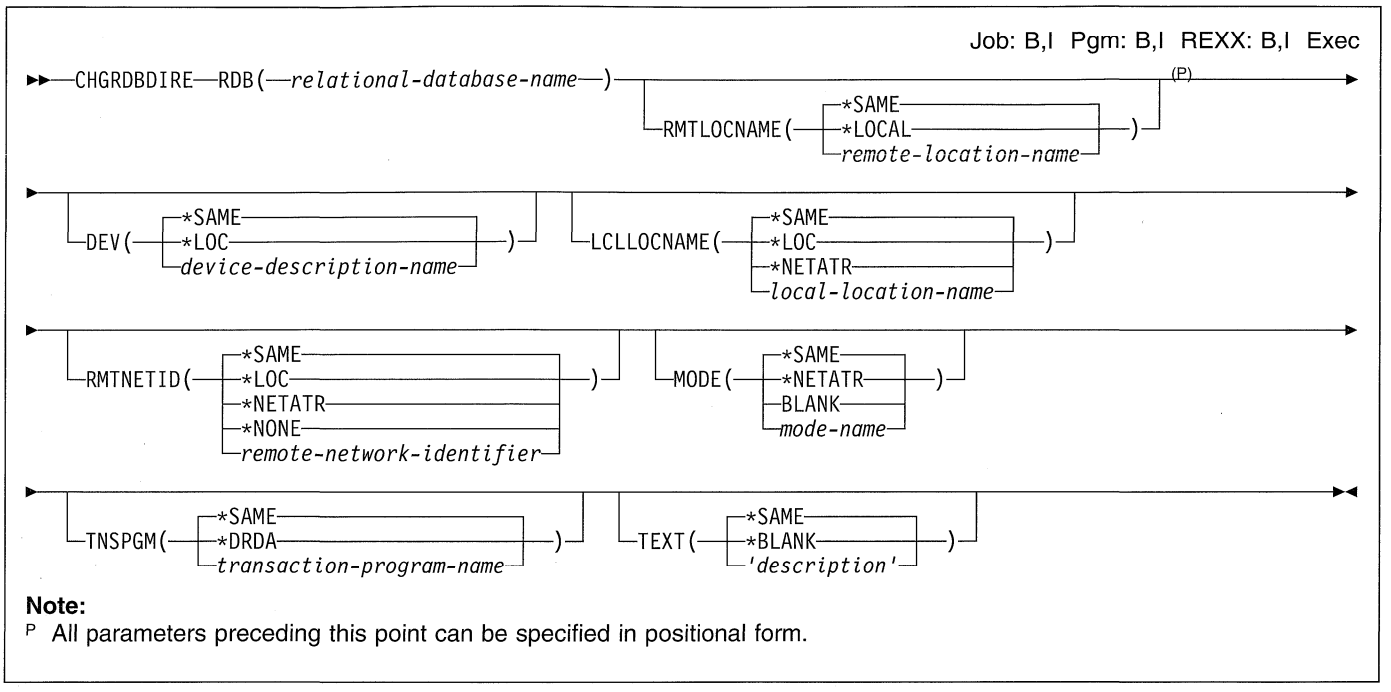
library-name: Specify the name of the library to be searched. If *SELECT is specified on the QSTDB parameter, any database in the library for which the user is authorized can be selected.

Example

```
CHGQSTDB
```

This command shows the Change Q & A Database display.

CHGRDBDIRE (Change Relational Database Directory Entry) Command



Note:
P All parameters preceding this point can be specified in positional form.

Purpose

The Change Relational Database Directory Entry (CHGRDBDIRE) command changes the values of an entry in the relational database directory. Values for any of the relational database's parameters, except its name, can be changed.

Note: Changes to an entry do not affect any connections that are using the relational database directory when the change is made. Changes take effect the next time a CONNECT operation is performed.

Required Parameter

RDB
Specifies the name of the relational database entry being changed. A maximum of 18 characters can be specified.

Optional Parameters

RMTLOCNAME
Specifies the remote location name of the system on which the relational database is located.

***SAME:** The remote location name does not change.

***LOCAL:** The relational database is located on the local system. *LOCAL can be specified for only one entry in the relational database directory since there is only one relational database on the local system.

remote-location-name: Specify a maximum of 8 characters for the remote location name.

DEV
Specifies the name of the advanced program-to-program communications (APPC) device description on the local system that is used with this relational database entry. The device description does not need to exist when the entry is changed.

More information is in the *APPC Programmer's Guide* and *APPN Guide*.

***SAME:** The device description does not change.

***LOC:** If APPC is being used, the system determines which device description is used. If advanced peer-to-peer networking (APPN) is being used, the system ignores this parameter.

device-description-name: Specify a maximum of 10 characters for the name of a device description.

LCLLOCNAME
Specifies the local location name by which this system is identified to the system on which the relational database is located. The local location name cannot be the same as the remote location name.

More information on local location names is in the *APPC Programmer's Guide*.

***SAME:** The local location name does not change.

***LOC:** If advanced program-to-program communications (APPC) is being used, the system determines which local location name is used. If advanced peer-to-peer networking (APPN) is being used, the system uses the default local location name defined in the network attributes.

***NETATR:** The local location name defined in the network attributes is used.

local-location-name: Specify a maximum of 8 characters for the local location name.

RMTNETID
Specifies the remote network identifier of the system on which the relational database is located.

***SAME:** The remote network identifier does not change.

***LOC:** If advanced program-to-program communications (APPC) is being used, the system determines which remote network identifier is used. If advanced peer-to-peer networking (APPN) is used, the system uses the local network identifier defined in this system's network attributes for the remote network identifier.

***NETATR:** The local network identifier defined in the local system's network attributes is used as the remote network identifier.

***NONE:** No remote network identifier is used.

remote-network-identifier: Specify a maximum of 8 characters for the remote network identifier.

More information on remote network identifiers is in the *APPC Programmer's Guide*.

MODE
Specifies the mode name to use with the remote location name to communicate with the system on which the relational database is located.

***SAME:** The mode name does not change.

***NETATR:** The mode name defined in the network attributes is used.

BLANK: A mode name of all blanks is used.

mode-name: Specify a maximum of 8 characters for the mode name.

More information on mode names is in the *APPC Programmer's Guide*.

TNSPGM
Specifies the name of the transaction program to use with the relational database entry.

***SAME:** The transaction program name does not change.

***DRDA:** The Distributed Relational Database Architecture (DRDA) transaction program name, X'07F6C4C2', is used. DRDA protocol is a means by which relational databases communicate with each other over a distributed network.

transaction-program-name: Specify the transaction program name in either of the following formats:

- A 4-byte hexadecimal name, which is entered by enclosing the 8 hexadecimal digits in apostrophes with a prefix of X. For example, X'07F6C4C2' is a 4-byte hexadecimal name.
- An 8-byte character name, which is entered by specifying the name in its 8-character form.

TEXT
Specifies text that briefly describes the relational database. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The text does not change.

***BLANK:** The text is changed to blanks.

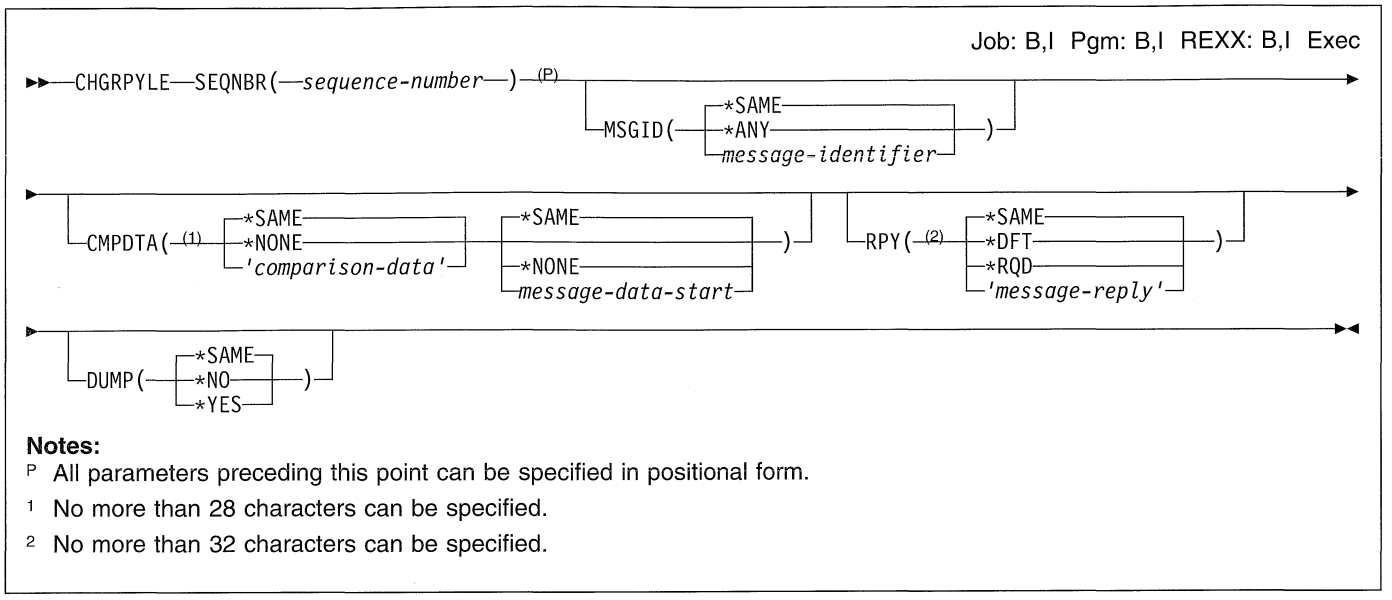
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGRDBDIRE   RDB(YOURRDB)  RMTLOCNAME(NEWARK)
```

This command changes a directory entry to use Newark as the new remote location name to access YOURRDB.

CHGRPYLE (Change Reply List Entry) Command



Notes:

P All parameters preceding this point can be specified in positional form.

1 No more than 28 characters can be specified.

2 No more than 32 characters can be specified.

Purpose

The Change Reply List Entry (CHGRPYLE) command changes a system reply list entry. Any of the attributes of a reply list entry may be changed, except for the sequence number.

The reply list is used as a source for automatic responses to predefined inquiry messages.

The reply list is only used when an inquiry message is *sent* by a job that has the system reply list attribute INQMSGRPY(*SYSRPYL) specified. The INQMSGRPY attribute can be changed with the CHGJOB command.

New entries may be added to the reply list with the Add Reply List Entry (ADDRPYLE) command; entries can be removed with the Remove Reply List Entry (RMVRPYLE) command. The entire list of entries can be shown with the Work with Reply List Entry (WRKRPYLE) command; from the display presented the user can add, change, and remove individual entries.

Restriction: This command is shipped with public *EXCLUDE authority and the QPGMR user profile has private authority to use the command.

Required Parameter

SEQNBR

Specifies the sequence number of the reply list entry being changed. The message identifier and message data of an inquiry message are matched against reply list entry message identifiers and comparison data in ascending sequence number order. The search ends when a match occurs or the last reply list entry is passed. Therefore, if more than one reply list entry matches the inquiry message identifier and comparison data, only the first entry that matches is used. If no reply list entry matches the inquiry message, the inquiry is sent, but no reply is sent unless a default reply would otherwise be sent; the job is not dumped.

Sequence numbers can range from 0001 through 9999. Duplicate sequence numbers are not allowed.

Optional Parameters

MSGID

Specifies the inquiry message identifiers for which automatic system action is taken. The message identifier may be specific or generic in scope. Only predefined messages (messages known to the system by a message identifier) can be matched by reply list entries; impromptu messages cannot be used for comparison.

If no comparison data is specified, then only the message identifier is used to match the message to this reply list entry. If this is the first entry in the message reply list which matches the message, then the action specified in this entry is taken.

***SAME:** The message identifier is not changed.

***ANY:** This reply list entry matches any message identifier. Unless this reply list entry has comparison data specified, any reply list entry with a higher sequence number than this one is ignored.

message-identifier: Specify a message identifier to compare with the message identifier of an inquiry message. The message identifier must be seven characters in length and in the format, pppnnnn.

The first three characters (ppp) must be a code consisting of one alphabetic character followed by two alphanumeric (alphabetic or decimal) characters. The

last four characters (nnnn) may consist of the decimal numbers 0 through 9 and the alphabetic characters A through F.

To specify a generic message identifier, enter zeros in the rightmost two or four positions of the numeric field, such as pppnn00 or ppp0000. For example, CPA0000 would match any CPA inquiry message, while CPA4200 would match any CPA42xx inquiry message.

CMPDTA

Specifies the comparison data that is used to determine whether this entry matches an inquiry message. This parameter is made up of comparison data and a start value for the message data. If the identifier of the inquiry message matches the message identifier of this reply list entry, then the message data specified for the inquiry message is compared to this data. If a message data start value has not been specified, then the first part of the message data (up through the first 28 characters or less) must exactly match the comparison data specified here before the action requested for this reply list entry is taken. However, if a start value has been specified, then the part of the message data beginning with the character position specified in the start value must exactly match the comparison data before any requested action is taken. If the comparison data is longer than the message data, then no match occurs. If no comparison data is specified, then only the message identifier is used to match the message to this reply list entry. If this is the first entry in the message reply list that matches the message, then the action specified in this entry is taken.

Message data for an inquiry message may be specified in the MSGDTA parameter of the Send User Message (SNDUSRMSG) and Send Program Message (SNDPGMMSG) commands for the inquiry message.

Element 1: Comparison Data

***SAME:** The value does not change.

***NONE:** No comparison data is specified. If the inquiry message has the specified identifier, the action specified by this reply list entry is taken.

'comparison-data': Specify a character string of no more than 28 characters (enclosed in apostrophes if blanks or other special characters are included). This string is compared with a string of the same length in the message data of the inquiry message, beginning with the first character (if no start value has been specified). If the comparison data string matches the inquiry message data string, the action specified by this reply list entry is taken.

Element 2: Message Data Start Position

***SAME:** The value does not change.

***NONE:** No message comparison data is specified.

message-data-start: Specify the character position in the message's replacement text (maximum value not to exceed 999) to be the point where the comparison data starts being compared with the replacement text. A start value is not valid without a specification of comparison data.

RPY

Specifies how to reply to an inquiry message that matches this reply list entry. The reply specified in this reply list entry is automatically sent by the system without requiring user intervention. The inquiry message does not interrupt or notify any job when the message arrives at the message queue. The inquiry message is not displayed before the reply is sent.

If a reply is not specified in this entry, a manual reply to the inquiry message must be made.

***SAME:** The value does not change.

***DFT:** The default reply to the inquiry message is sent. If no default reply is specified in the message description of the inquiry message, the system default reply, *N, is used.

***RQD:** The inquiry message requires an explicit reply. If the message queue to which the inquiry is sent is in break mode, the message interrupts the job and is shown on the display. If the message queue is in notify mode, the job to which it is allocated is notified. No reply is automatically sent.

'message-reply': Specify a character string of up to 32 characters (enclosed in apostrophes if blanks or other special characters are included), which is sent as a reply to the inquiry message. If this reply is not valid for the inquiry message, the inquiry is sent if RPY(*RQD) has been specified.

DUMP

Specifies whether the contents of the job that sent the inquiry message are printed (dumped) when the inquiry message matches this reply list entry. The dump is the same as the dump specified by DMPLST(*JOB) on the Add Message Description (ADDMSGD) command or by the command DSPJOB OUTPUT(*PRINT) for the sending job. A job dump may be requested regardless of the value specified for the RPY parameter.

***SAME:** The value does not change.

***NO:** The job is not dumped.

***YES:** The job is dumped before control returns to the program sending the message.

Examples

Example 1: Changing the Message Identifier

```
CHGRPYLE   SEQNBR(20)  MSGID(RPG1299)
```

This command changes the message identifier of the reply list entry (sequence number 20) to RPG1299. Whenever an RPG1299 inquiry message is sent by a job that is using the reply list, the action previously specified for entry 20 is taken.

Example 2: Changing the Comparison Data

```
CHGRPYLE    SEQNBR(25)   CMPDTA(MYPROGRAM)
```

This command changes the comparison data of the reply list entry whose sequence number is 25 to MYPROGRAM. This entry only matches inquiry messages whose message data begins with MYPROGRAM. For example, if this entry were for the RPG1200 messages, the entry is used only when the RPG program from which the message was sent has message data named MYPROGRAM.

Example 3: Changing the Reply Sent

```
CHGRPYLE    SEQNBR(30)   RPY(C)
```

This command changes the reply sent for the reply list entry whose sequence number is 30 to C. Whenever an inquiry message which matches the message identifier and comparison data previously defined for this entry is sent by a job that is using the reply list, a 'C' reply is automatically sent.

Example 4: Printing the Job Contents

```
CHGRPYLE    SEQNBR(40)   DUMP(*YES)
```

This command changes the attribute defined for the DUMP parameter for the reply list entry whose sequence number is 40. Whenever this entry matches an inquiry message, the sending job is dumped before control returns to the sending program.

Example 5: Sending a Manual Reply

```
CHGRPYLE    SEQNBR(45)   MSGID(CPA5300)   CMPDTA(*NONE)
  RPY(*RQD)   DUMP(*NO)
```

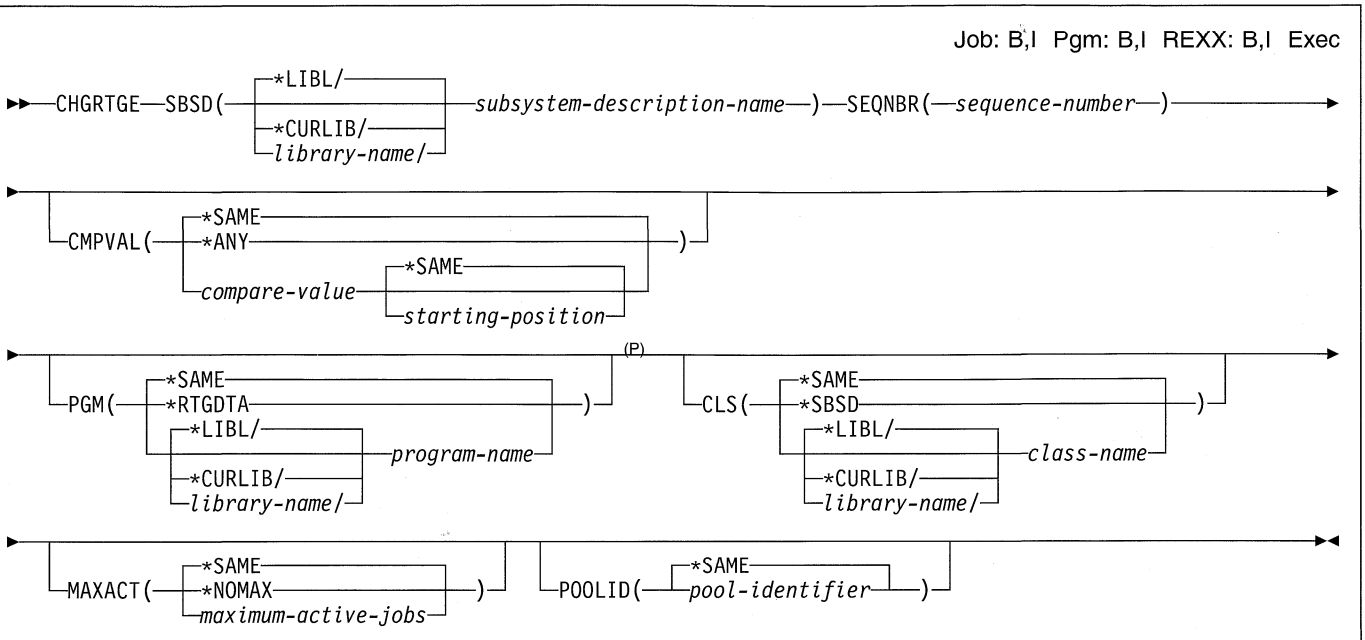
This command changes some of the attributes of the reply list entry whose sequence number is 45. Whenever a CPA53xx inquiry message is sent by a job that is using the reply list, a manual reply must be issued. If the message queue to which the inquiry is sent is in break mode, the message interrupts the job. The sending job is not dumped.

Example 6: Sending an Automatic Reply

```
CHGRPYLE    SEQNBR(9999)   MSGID(CPA3917)   RPY(R)
```

This command changes the reply list entry whose sequence number is 9999. Whenever a CPA3917 inquiry message is sent by a job that is using the reply list, an 'R' reply is automatically sent. The inquiry does not break into the message queue, and no opportunity is given to reply to the message manually.

CHGRTGE (Change Routing Entry) Command



Note:

| P All parameters preceding this point can be specified in positional form.

Purpose

The Change Routing Entry (CHGRTGE) command changes a routing entry in the specified subsystem description. The associated subsystem must be inactive when the changes are made. The routing entry specifies the parameters used to start a routing step for a job.

Restriction: To use this command, the user must have object operational and object management authorities for the subsystem description being changed.

Required Parameters

SBSD

Specifies the qualified name of the subsystem description containing the routing entry being changed.

| The name of the subsystem description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

subsystem-description-name: Specify the name of the subsystem description that contains the routing entry.

SEQNBR

Specifies the sequence number of the routing entry that is added or changed. Specify the sequence number, ranging from 1 through 9999, of the routing entry.

Optional Parameters

CMPVAL

Specifies a value that is compared with the routing data to determine whether this is the routing entry to be used for starting a routing step. Optionally, a new starting position in the routing data character string can be specified for the comparison. If CMPVAL is not specified, *SAME is assumed. If a starting position value is not specified, *SAME is assumed.

Element 1: Compare Value

***SAME:** The value does not change.

***ANY:** Any routing data is considered a match. To specify *ANY, this routing entry must also have the highest SEQNBR value of any routing entry in the subsystem description.

compare-value: Specify a new value (a character string not exceeding 80 characters) that is compared with routing data for a match. When a match occurs, this routing entry is used to start a routing step. A starting position in the routing data character string can be specified for the comparison. If no position is specified, *SAME is assumed.

Element 2: Starting Position

***SAME:** The value does not change.

starting-position: Specify a value indicating which position in the routing data character string is the starting position for the comparison. The last character position compared must be less than or equal to the length of the routing data used in the comparison.

PGM

Specifies the qualified name of the program called as the (first) program run in the routing step. No parameters can be passed to the specified program. The program name can either be explicitly specified in the routing entry or extracted from the routing data. If a program name is specified in a routing entry, selection of that routing entry results in the routing entry program being called (regardless of the program name passed in an EVOKE request). If the program specified in the EVOKE request is called, PGM(*RTGDTA) must be specified in the routing entry.

***SAME:** The value does not change.

***RTGDTA:** The program name is taken from the routing data supplied and matched against this entry. A program name is taken from the routing data in the following manner: the program name is extracted from positions 37 through 46, and the library name is extracted from positions 47 through 56. Care should be used to ensure that routing entries that specify *RTGDTA are selected only for EVOKE requests on jobs that have specified the program name in the correct position in the routing data.

The name of the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the qualified name of the program called and run in the routing step. If the program does not exist when this routing entry is changed, a library qualifier must be specified because the program name is retained in the subsystem description.

CLS

Specifies the qualified name of the class used for the routing steps started through this routing entry. The class defines the attributes of the running environment for processing the routing step associated with this routing entry. If the class does not exist when this routing entry is changed, a library qualifier must be specified because the class name is retained in the subsystem description. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***SBSD:** The class having the same name as the subsystem description, specified by the SBSD parameter, is used for routing steps started through this entry.

The name of the class can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

class-name: Specify the qualified name of the class used for routing steps started through this routing entry.

MAXACT

Specifies the maximum number of routing steps (jobs) that can be active at the same time through this routing entry. In a job, only one routing step is active at a time. When a subsystem is active and the maximum number of routing steps is reached, any subsequent attempts to start a routing step through this routing entry fails. If the routing data is entered interactively, an error message is sent to the user. Otherwise, the job is ended and a message is sent by the subsystem to the job's log. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***NOMAX:** There is no maximum number of jobs that can be active at the same time.

maximum-active-jobs: Specify a value for the new maximum number of routing steps that can be active at the same time through this routing entry. If a routing step would exceed this number if it were started, the job is implicitly ended.

POOLID

Specifies the pool identifier of the storage pool in which the program runs. The pool identifier specified here relates to the storage pools in the subsystem description.

***SAME:** The value does not change.

pool-identifier: Specify the identifier of another existing storage pool in which the routing step is to run. Valid values range from 1 through 10.

Examples

Example 1: Changing Class and Pool ID

```
CHGRTGE  SBSD(LIB5/ORDER)  SEQNBR(1478)  CLS(LIB6/SOFAST)
  POOLID(3)
```

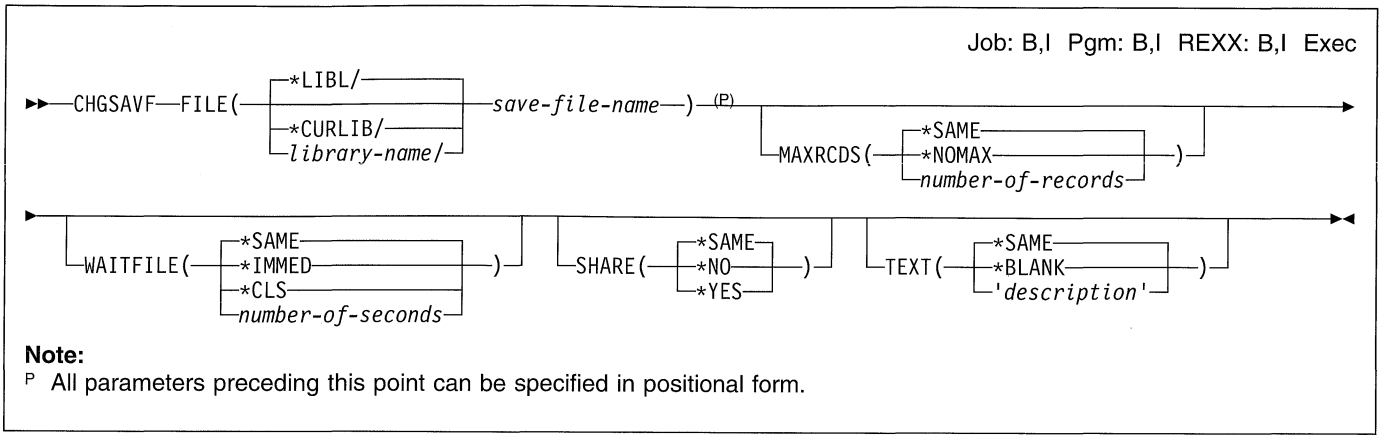
This command changes routing entry 1478 in the subsystem description ORDER found in library LIB5. The same program is used, but now it runs in storage pool 3 using class SOFAST in library LIB6.

Example 2: Changing the Name of the Program Called

```
CHGRTGE  SBSD(T7/PGMR)  SEQNBR(157)  PGM(T7/INTDEV)
```

This command changes routing entry 157 in the subsystem description PGMR found in library T7. The program INTDEV in library T7 is now called whenever this routing entry is selected. The other routing entry parameters are not changed.

CHGSAVF (Change Save File) Command



Note:
P All parameters preceding this point can be specified in positional form.

Purpose

The Change Save File (CHGSAVF) command changes the attributes of a specified save file. The changes become a permanent part of the file and are kept until the file is either changed or deleted.

Restrictions: The user of this command must have: 1) object operational authority and object management authority for the save file, and 2) read authority for the specified library.

Required Parameter

FILE

Specifies the qualified name of the save file whose attributes are changed.

The name of the save file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

save-file-name: Specify the name of the file that is to be used.

Optional Parameters

MAXRCDS

Specifies the maximum number of records the save file can reach. The size of the save file is estimated in bytes at about 8192 + (512 × number of records in the save file).

There is room for approximately two thousand 512-byte records in 1 megabyte of space. For example, to ensure that the save file does not exceed approximately 20 megabytes (20 x 2000), specify MAXRCDS (40000).

If the current number of records in the save file is greater than the new MAXRCDS, an error message is sent, the command ends, and the save file is not changed.

***SAME:** The value does not change.

***NOMAX:** The system maximum is used.

number-of-records: Specify the maximum number of records that the save file can contain. Valid values range from 1 through 5865293.

WAITFILE

Specifies the number of seconds that the program waits for the file resources and session resources to be allocated when the file is opened, or for the device or session resources to be allocated when an acquire operation is performed to the file. If those resources are not allocated within the specified wait time, an error message is sent to the program. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

Note: An immediate allocation of the device by the device resource is required when an acquire operation is performed to the file.

***SAME:** The value does not change.

***IMMED:** The program does not wait; when the file is opened, an immediate allocation of the file resources is required.

***CLS:** The job default wait time is used as the wait time for the file resources being allocated.

number-of-seconds: Specify the number of seconds that the program waits for the file resources to be allocated. Valid values range from 1 through 32767 seconds.

SHARE

Specifies whether the open data path (ODP) for the save file is shared with other programs in the routing step.

When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

More information on shared database files is in the *Database Guide*.

***SAME:** The value does not change.

***NO:** The ODP created by the program with this attribute is not shared with other programs in the routing step. Every time a program opens the file with this attribute, a new ODP to the file is created and activated.

Note: This includes multiple opens in the same program.

***YES:** The ODP created with this attribute is shared with each program in the routing step that also specifies SHARE(*YES) when it opens the file.

Note: When SHARE(*YES) is specified and control is passed to a program, a read operation in that program retrieves the next input record. A write operation produces the next output record.

TEXT

Specifies text that briefly describes the save file. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Examples

Example 1: File Resources Allocated Immediately

```
CHGSAVF   FILE(ONLINE)  WAITFILE(*IMMED)
```

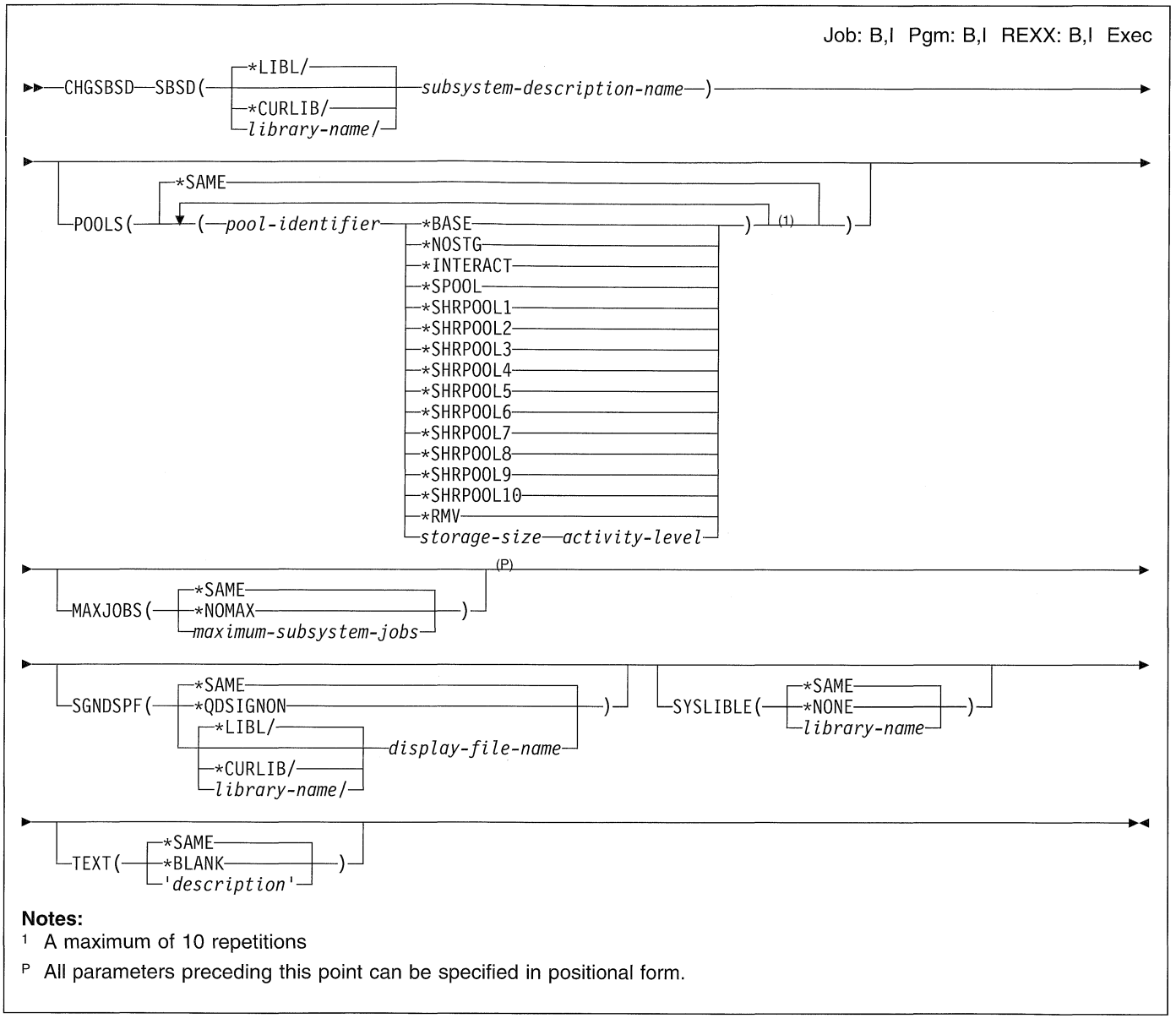
This command changes the save file named ONLINE so that when it is opened the file resources must be available immediately, or an error message is sent. No other files are changed.

Example 2: Changing Maximum Number of Records

```
CHGSAVF   FILE(ONLINE)  MAXRCDS(20000)
```

This command changes the save file named ONLINE so that it can have up to 20,000 records (approximately 10 megabytes).

CHGSBSD (Change Subsystem Description) Command



Notes:

1 A maximum of 10 repetitions

P All parameters preceding this point can be specified in positional form.

Purpose

The Change Subsystem Description (CHGSBSD) command changes the operational attributes of the specified subsystem description. This command can be sent while the subsystem is active. Changing the POOLS parameter might require the subsystem to be inactive.

Restriction: The user of this command must have object operational authority and object management authority for the subsystem description before it can be changed. The user must also have object operational authority for the library containing the subsystem description.

Required Parameter

SBSD

Specifies the qualified name of the subsystem description to which changes are made.

The name of the subsystem description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

subsystem-description-name: Specify the name of the subsystem description being changed.

The name of the IBM-supplied controlling subsystem, QCTL, must not be specified for this parameter. The CRTSBSD command, however, can be used to create a similar subsystem description, and the user can specify it as the controlling subsystem by specifying its name in the QCTLSBSD system value instead of the QCTL library.

Optional Parameters

POOLS

Specifies the identifiers of one or more storage pool definitions and the changes that are made to them. When a new storage pool definition is added or an existing pool definition is removed, the subsystem must be inactive. The size and activity level of each existing pool definition that is not specified does not change. More information about storage pools is in the *Advanced Backup and Recovery Guide* and the *Work Management Guide.*

***SAME:** The value does not change.

Element 1: Pool Identifier

pool-identifier: Specify the pool identifier, ranging from 1 through 10, of the storage pool definition added or deleted, or whose attributes are changed. If more than one pool definition is specified, the attributes of each definition must follow its identifier.

Element 2: Pool Definition Options

***BASE:** The specified pool definition is the system base pool, which can be shared with other subsystems. The size and activity level of the shared system pool are specified in the system values, QBASPOOL and QBASACTLVL.

***NOSTG:** No storage and no activity level are assigned to the pool at the present time.

***INTERACT:** The shared pool used for interactive work.

***SPOOL:** The shared pool used for spooled writers.

***SHRPOOL1:** A general purpose shared pool.

***SHRPOOL2:** A general purpose shared pool.

***SHRPOOL3:** A general purpose shared pool.

***SHRPOOL4:** A general purpose shared pool.

***SHRPOOL5:** A general purpose shared pool.

***SHRPOOL6:** A general purpose shared pool.

***SHRPOOL7:** A general purpose shared pool.

***SHRPOOL8:** A general purpose shared pool.

***SHRPOOL9:** A general purpose shared pool.

***SHRPOOL10:** A general purpose shared pool.

***RMV:** The specified pool definition is removed from the document.

storage-size activity-level: Specify the storage size (in kilobytes) of the specified storage pool. The storage pool identifier and size must be specified. A value of at least 16 kilobytes must be specified for the storage size. For activity level, specify the maximum number of jobs that can run in the pool at the same time. Active jobs running in a pool must finish before the pool can be deallocated by the system. When this command is used to change a pool size allocation, the existing jobs in the pool continue to process until they have finished.

If the subsystem is active, and if a save and/or restore operation is using some of the storage allocated to the specified pool, a request to reduce a pool allocation may not take effect immediately. In that case, the pool size allocation is changed when the machine cancels the storage being used. Use the Display System Status (DSPSYSSTS) command to determine the amount of storage reserved for save/restore operations.

MAXJOBS

Specifies the maximum number of jobs that can be active at the same time in the subsystem controlled by this subsystem description. The maximum applies to all jobs that are started and are waiting or running, but not to jobs on the job queue or jobs that have finished running.

***SAME:** The value does not change.

***NOMAX:** There is no maximum number of jobs that can be active at the same time.

maximum-subsystem-jobs: Specify the maximum number of active jobs allowed in this subsystem.

SGNDSPF

Specifies the qualified name of the sign-on display file that is used when showing sign-on displays at work stations allocated to the subsystem. If the specified sign-on display file does not exist when the subsystem description is changed, the user must specify a library qualifier because the sign-on display file name is kept by the system. The sign-on display file must contain a record format named SIGNON. More information on display files is in the *Guide to Programming Displays.*

Note: The sign-on display file can be changed when the subsystem is active. However, the new sign-on display file is not used until the next time the subsystem is started.

***SAME:** The value does not change.

***QDSIGNON:** The sign-on display file QDSIGNON in QSYS is used when showing sign-on displays at work stations that are allocated to the subsystem.

The name of the sign-on display file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

If the file is found, its library name is placed in the subsystem description.

display-file-name: Specify the name of the sign-on display file used.

SYSLIBLE

Specifies a library that is specified ahead of other libraries on the system library list. This parameter allows the user to change a secondary language library, causing messages and displays to appear in the user's spoken language.

Note: This parameter can be used while the subsystem is active. Any changes the user makes to this parameter take effect for new jobs that are started. The library list of active jobs within the subsystem is not changed.

***SAME:** The value does not change.

***NONE:** The secondary language library is removed from the system library list.

library-name: Specify the name of the library being added to the system library list.

TEXT

Specifies text that briefly describes the subsystem description. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

Note: The text can be changed only when the subsystem description is not active.

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text enclosed in apostrophes.

Examples

Example 1: Changing Storage Size and Activity Level

```
CHGSBSD  SBSD(QGPL/PAYCTL)  POOLS((2 150 3))
  SGNDSPF(QGPL/COMPANYA)
```

This command changes the definition of storage pool 2 that is used by subsystem PAYCTL to a storage size of 150 K and an activity level of 3. The sign-on display file is changed to display file COMPANYA and is located in the QGPL library. If the subsystem is active when this command is issued, COMPANYA is not used until the next time the subsystem is started.

Example 2: Changing Multiple Attributes

```
CHGSBSD  SBSD(LIB6/ORDER)
  POOLS((1 *BASE)(2 75 4)(3 *RMV)(4  *NOSTG))
  MAXJOBS(5)
```

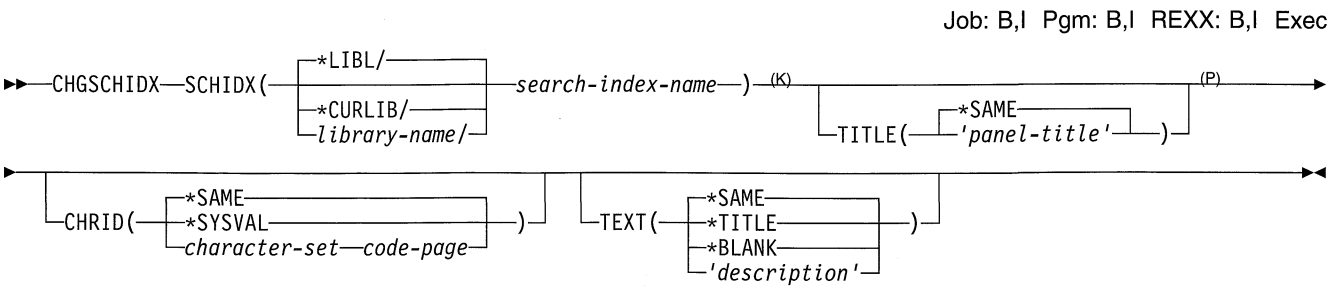
This command changes the maximum number of jobs that subsystem ORDER can support to five. (The description of the subsystem is stored in library LIB6.) The definition of storage pool 1 is changed to the shared system pool, the definition of pool 2 is changed to have a storage size of 75K and an activity level of 4, the definition of pool 3 is removed from the subsystem, and the definition of pool 4 is changed to have no storage and no activity level.

Example 3: Changing the Language Library

```
CHGSBSD  SBSD(QGPL/SPANISH)
  SGNDSPF(QSYS2931/QDSIGNON)
  SYSLIBLE(QSYS2931)
```

This command changes subsystem description SPANISH to a Spanish secondary language.

CHGSCHIDX (Change Search Index) Command



Notes:

K All parameters preceding this point are key parameters.

P All parameters preceding this point can be specified in positional form.

Required Parameter

The Change Search Index (CHGSCHIDX) command is used to change a search index. A search index refers to help information contained in one or more panel groups. The user can access help information panel groups through data description specifications (DDS) by pressing the HELP key, or through the information search function.

Restriction: The user must have *CHANGE authority for the search index that is being changed.

Required Parameters

SCHIDX

Specifies the qualified name of the search index being changed.

The name of the search index can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

search-index-name: Specify the name of the search index being changed.

Optional Parameters

TITLE

Specifies the title to appear at the top of the panel on which the search information is presented.

***SAME:** The title of the search index does not change.

'panel-title': Specify no more than 55 characters of text, enclosed in apostrophes.

CHRID

Specifies the graphic character set and code page values to be used for the search index. This value can be changed only when there are no entries in the search index. The value for this parameter must match the TXTCHRID value of panel groups added to this search index.

***SAME:** The character set and code page values do not change.

***SYSVAL:** The QCHRID system value contains the values to be used to interpret search index values.

Element 1: Character Set

character-set: Specify the graphic character set values that match the code page of the synonyms used in the search index.

Element 2: Code Page

code-page: Specify the code page values that match the code page of the synonyms used in the search index.

TEXT

Specifies text that briefly describes the search index. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The text does not change.

***TITLE:** The first 50 characters of the title are used as the title for the search index.

***BLANK:** No text is specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

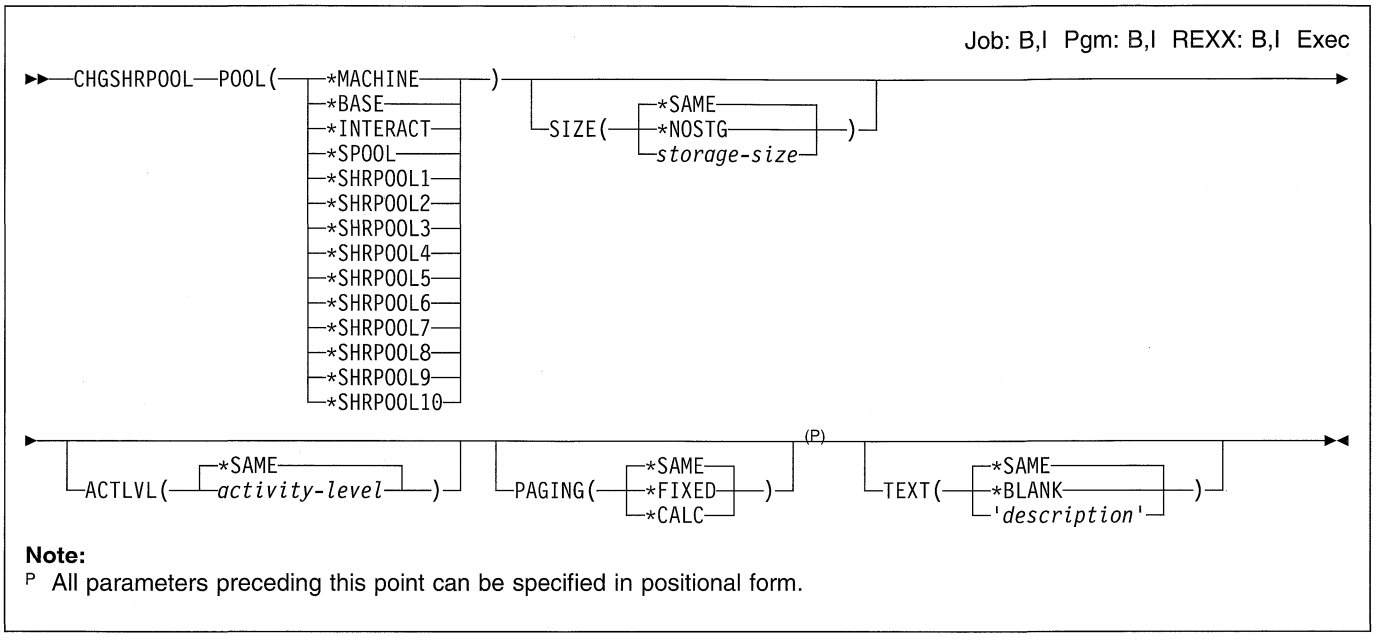
Example

```
CHGSCHIDX   SCHIDX(ACCOUNTING)
  TITLE('Accounting Help Index')
  TEXT('Accounting Help Index')
```

CHGSCHIDX

This command changes the search index ACCOUNTING in the current library.

CHGSHRPOOL (Change Shared Storage Pool) Command



Note:

P All parameters preceding this point can be specified in positional form.

Purpose

The Change Shared Storage Pool (CHGSHRPOOL) command changes the size or activity level for a shared pool. If the user has the authority to use this command, then a shared pool can be changed. The change to a shared pool takes effect immediately if the shared pool is active, that is, in use by a subsystem or active job. The immediate effect also depends on whether storage is available at the time the shared pool is changed. If a shared pool is not active, the change takes effect as soon as a subsystem is started using a shared pool.

Required Parameter

POOL

Specifies the shared storage pool being changed. More information about storage pools is in the *Advanced Backup and Recovery Guide* and the *Work Management Guide*.

***MACHINE:** The machine pool used for Licensed Internal Code. Only the size can be changed for the machine pool. This is the same as using the Change System Value (CHGSYSVAL) command to change the QMCHPOOL system value.

***BASE:** The base pool. Only the activity level can be changed for the base pool. This is the same as using the Change System Value (CHGSYSVAL) command to change the QBASACTLVL system value.

***INTERACT:** The shared pool for interactive work.

***SPOOL:** The shared pool used for spooled writers.

***SHRPOOL1:** A general purpose shared pool.

***SHRPOOL2:** A general purpose shared pool.

***SHRPOOL3:** A general purpose shared pool.

***SHRPOOL4:** A general purpose shared pool.

***SHRPOOL5:** A general purpose shared pool.

***SHRPOOL6:** A general purpose shared pool.

***SHRPOOL7:** A general purpose shared pool.

***SHRPOOL8:** A general purpose shared pool.

***SHRPOOL9:** A general purpose shared pool.

***SHRPOOL10:** A general purpose shared pool.

Optional Parameters

SIZE

Specifies the size of the storage pool expressed in kilobytes (1K= 1024 bytes). This is the amount of main storage that can be used by a pool. A value of at least 32 (32 K) must be specified for the storage size.

***SAME:** The value does not change.

***NOSTG:** No storage or activity level is defined for the pool. A pool cannot be changed to *NOSTG if it is being used by an active subsystem or active job or if the pool has reserved storage.

storage-size: Specify the size of the storage pool.

ACTLVL

Specifies the maximum number of jobs that can run at the same time in a pool.

***SAME:** The value does not change.

activity-level: Specify the activity level for the pool.

PAGING

Specifies the paging option associated with the pool. The paging option determines whether the system dynamically adjusts the paging characteristics for the storage pool for optimum performance.

***SAME:** The value does not change.

***FIXED:** The system default values are used. The system does not dynamically adjust the paging characteristics for the storage pool.

***CALC:** The system dynamically adjusts the paging characteristics for the storage pool to ensure optimum performance.

TEXT

Specifies text that briefly describes the shared pool. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

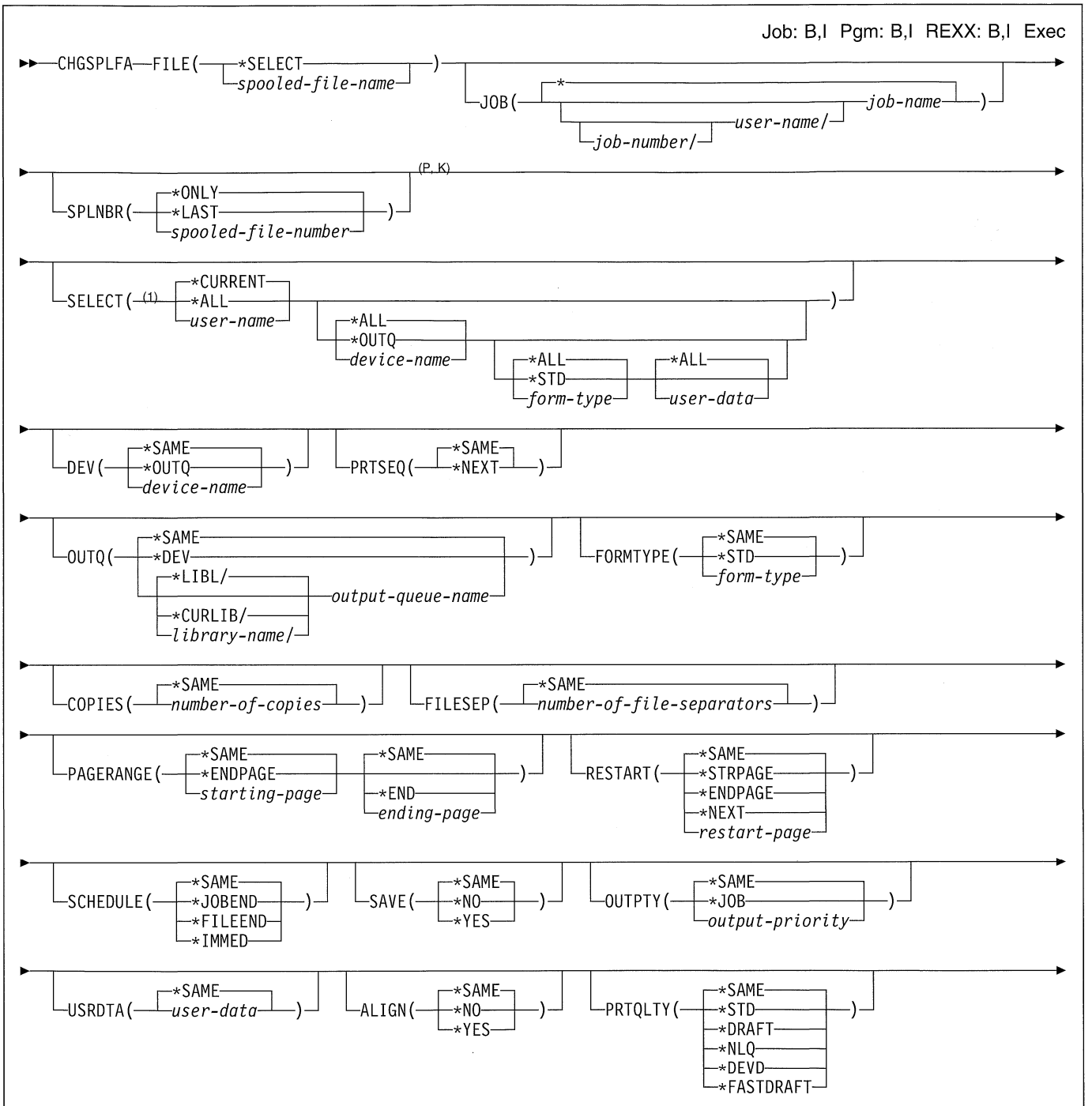
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGSHRPOOL  POOL(*INTERACT)  SIZE(4200)  ACTLVL(*SAME)
  PAGING(*SAME)
```

This command changes the size of the interactive pool to 4200 Kilobytes. The activity level and paging option remains the same.

CHGSPLFA (Change Spooled File Attributes) Command

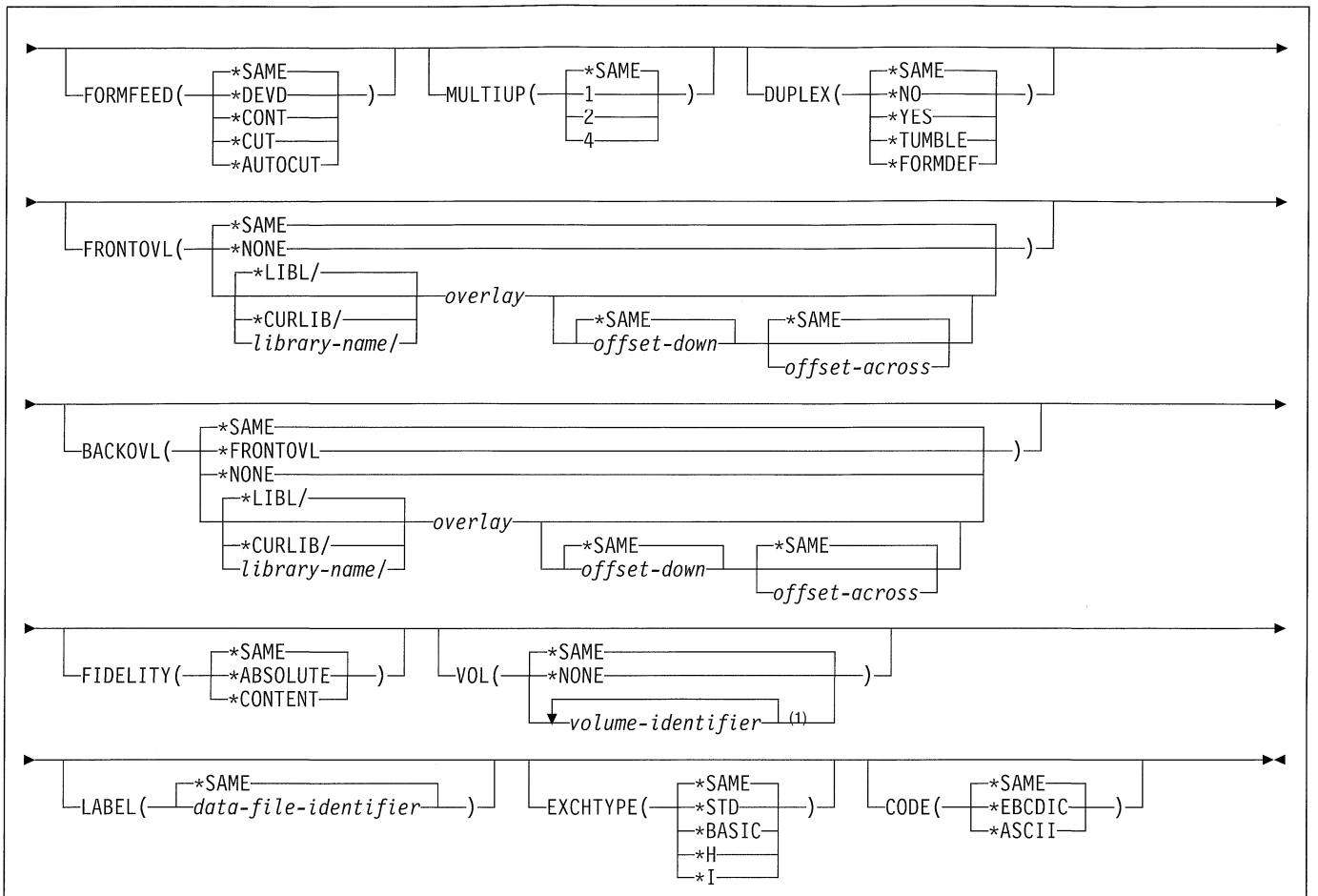


Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

1 SELECT only used if FILE(*SELECT) is specified.



Note:
1 A maximum of 50 repetitions

Purpose

The Change Spooled File Attributes (CHGSPLFA) command changes the attributes of a spooled file while it is on an output queue. The changes affect only the current processing of the file. The next time the job runs and the file is produced, the file attributes are derived from the device file description, the program, and any override commands.

If the file is currently being produced on an output device, the only parameters that can be changed are COPIES, RESTART, and SAVE. An attempt to change any other parameter results in an error, and no file attributes are changed. However, if the file is being held on an output queue because of spooling attribute errors, this command can be used to change the attributes, and a spooling writer can then be started to produce the file.

Required Parameter

FILE

Specifies the name of the spooled file whose attributes are being changed.

***SELECT:** All spooled files that meet the selection values specified in the SELECT keyword are changed. This value is mutually exclusive with the JOB and SPLNBR parameters.

spooled-file-name: Specify the name of the spooled file. If *SELECT is specified, the JOB and SPLNBR keywords are ignored.

Optional Parameters

JOB

Specifies the name of the job that created the spooled file.

A job identifier is a qualified name with up to three elements. For example:

```
job-name
user-name/job-name
job-number/user-name/job-name
```

| More information on this parameter is in Appendix A,
| "Expanded Parameter Descriptions."

*: The job that issued this CHGSPLFA command is the job that created the spooled file.

If no job qualifier is given, the jobs currently in the system are searched for the simple job name.

job-name: Specify the name of the job that created the spooled file.

user-name: Specify the specific or generic user name of the job to omit. Because jobs may have identical user names, this value may not identify a specific job. OMTUSRID(user-name) is equivalent to OMTJOB(*N/user-name/*N).

job-number: Specify the number of the job that created the spooled file.

SPLNBR

Specifies the number of the spooled file in the job whose attributes are being changed. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***ONLY:** One spooled file from the job has the specified file name. The number of the spooled file is not necessary. If *ONLY is specified and more than one spooled file has the specified file name, a message is sent.

***LAST:** The spooled file with the highest number and the specified file name is used.

spooled-file-number: Specify the number of the spooled file having the specified file name whose attributes are being changed.

SELECT

Specifies which group of files have their attributes changed. Files can be selected based on user, device, form type, and user data. Only files that meet each of the specified requirements are selected.

Element 1: User Values

***CURRENT:** Only files created by the user running this command are selected.

***ALL:** Files created by all users or on any output queue are selected.

user-name: Specify only files created by the named user.

Element 2: Device Values

***ALL:** Files queued for any device are selected.

***OUTQ:** All files not queued for a device are selected. These files are on output queues not associated with printers.

device-name: Specify only files queued for the specified device name.

Element 3: Form Type Values

***ALL:** Files for all form types are selected.

***STD:** Only files that specify the standard form type are selected.

form-type: Specify only files with the specified form type.

Element 4: User Data Values

***ALL:** Files with a user data tag specified are selected.

user-data: Specify only files with the specified user data tag.

DEV

Specifies the name of the printer on which the file is printed.

***SAME:** The current value does not change.

***OUTQ:** The file is not assigned to a specific printer. It is placed on the output queue specified on the OUTQ parameter.

device-name: Specify the name of the printer that prints the file.

PRTSEQ

Specifies whether the file is next on the output queue to be printed. To move the file to the top of the queue, the following steps are taken:

1. If the file is held, it is released
2. The SCHEDULE attribute changes to *IMMED
3. The output priority (OUTPTY) changes to a priority that places it at the top of the queue

Because of these actions, PRTSEQ is mutually exclusive with the SCHEDULE and OUTPTY parameters.

***SAME:** The value does not change.

***NEXT:** The attributes of the file are changed to move it to the top of the output queue. If PRTSEQ(*NEXT) is specified when the SELECT parameter is specified, files with selection values are moved ahead of the files that do not meet the requirements. Two files that are both moved may change their relative positions on the output queue.

OUTQ

Specifies the qualified name of the output queue.

Note: If the spooled file is currently on an output queue for which DSPDTA(*OWNER) was specified on the Create Output Queue (CRTOUTQ) or Change Output Queue (CHGOUTQ) command, the user must own the file or have *SPLCTL authority to move it.

***SAME:** The value does not change.

***DEV:** The output queue specified on the PRTDEV parameter is used.

The name of the output queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

output-queue-name: Specify the name of the output queue to which the file is moved.

FORMTYPE

Specifies the type of form on which the output is printed. The identifiers used to indicate the type of forms are user-defined and can be a maximum of 10 characters in length.

Note: *SAME must be specified if the output file is a diskette file.

***SAME:** The value does not change.

***STD:** The standard form type is used. The output is printed on the form type specified in the printer file for the printers selected. The printer file contains information that controls how the document is printed on a particular printer.

form-type: Specify the forms identifier for the forms used to produce the output of this spooled file. Up to 10 alphanumeric characters can be specified. Strings with embedded blanks must be enclosed in apostrophes.

COPIES

Specifies, for spooled files, the number of copies being printed.

Note: Specifying a value while a file is being printed causes the specified number of copies to print in addition to the number of copies that have already been printed.

***SAME:** The value does not change.

number-of-copies: Specify the number of copies to print. Valid values range from 1 through 255.

FILESEP

Specifies the number of separator pages to produce at the beginning of each printed output file to separate the file from the other files being spooled to an output device. The identifying information included on each file separator is the file name and number, the job name and number, and the user's name. *SAME must be specified if the spooled file is a diskette file.

***SAME:** The value does not change.

number-of-file-separators: Specify the number of pages, ranging from 0 through 9, used as file separators.

PAGERANGE

Specifies the page range to print for each copy of the file to be printed.

Element 1: Starting Page to Print

***SAME:** The value does not change.

***ENDPAGE:** The page on which to start printing is the same as the page on which to end printing.

starting-page: Specify the page on which to start printing.

Element 2: Ending Page to Print

***SAME:** The value does not change.

***END:** The last page in the file is printed.

ending-page: Specify the page on which to end printing.

RESTART

Specifies the page on which to restart printing. Specifying a value while a file is being printed causes the printer writer to stop printing the file and restart on the page specified by the user. If the file is not being printed, the change occurs when the first copy is printed. After the printer writer is repositioned to the page specified on this parameter, the value is reset to RESTART(*STRPAGE).

***SAME:** The value does not change.

***STRPAGE:** The starting page specified on the PAGERANGE parameter is the page on which printing is restarted. This value is the default value that is used when a spooled file is created. Also, RESTART is reset to *STRPAGE after the printer writer repositions to the page specified on this parameter.

***ENDPAGE:** The ending page is the page on which to restart printing. Therefore, only the ending page is printed.

***NEXT:** This value is valid only when a spooled file has been interrupted while it is being printed by the HLDSPLF command where OPTION(*PAGEEND) is specified or the ENDWTR command where OPTION(*PAGEEND) is specified.

restart-page: Specify the page on which to restart printing.

SCHEDULE

Specifies when the output file is available to the writer.

***SAME:** The value does not change.

***JOBEND:** The spooled file is available to the writer only after the entire job is completed.

***FILEEND:** The spooled file is available to the writer as soon as the file is closed in the program.

***IMMED:** The spooled file is made available to the writer as soon as the file is opened in the program.

SAVE

Specifies whether the spooled file is saved after it has been written to an output device.

***SAME:** The value does not change.

***NO:** The spooled file data is not retained on the output queue after it has been produced.

***YES:** The spooled file data is retained on the output queue until the file is deleted. After the file is produced, the number of copies (COPIES parameter) is set to 1, and the status of the file is changed from WTR to SAV. Refer to the RLSSPLF command description for information on how to produce the spooled file.

OUTPTY

Specifies the output priority assigned to the spooled file.

***SAME:** The value does not change.

***JOB:** The output priority associated with the job that created the spooled file is used.

output-priority: Specify the assigned output priority. Valid values range from 1 (high priority) through 9 (low priority).

USRDTA

Specifies, for spooled output only, the user-specified data that identifies the file.

***SAME:** The value does not change.

user-data: Specify up to 10 characters of data assigned to the spooled file.

ALIGN

Specifies whether to verify forms alignment on this file. This parameter is used only by print writers which were started with the ALIGN(*FILE) option, and can be changed only on spooled files which are to be printed. *SAME must be specified for diskette files.

***SAME:** The value does not change.

***NO:** The forms alignment is not verified.

***YES:** The forms alignment is verified.

PRTQLTY

Specifies the quality of the print produced. Refer to the CRTPRTF (Create Printer File) command description for a list of printers supporting the PRTQLTY parameter.

Note: The user can also specify PRTQLTY(*NLQ) for the 5219 and 4214 Printers to get the best print quality.

***SAME:** The value does not change.

***STD:** The output is printed with standard quality.

***DRAFT:** The output is printed with draft print quality.

***NLQ:** The output is printed with near letter quality.

***DEVD:** The output is printed with the default print quality for the printer.

***FASTDRAFT:** The output is printed at a higher speed and with lower quality than it would be if you specified *DRAFT.

FORMFEED

Specifies, for the 4214, 5219, and 5553 printers (including ASCII printers that are configured as an SCS 4214 or SCS 5219 printer), and for IPDS printers, the mode in which forms are fed into the device.

***SAME:** The value does not change.

***DEVD:** The forms are fed into the printer in the manner specified in the device description.

***CONT:** Continuous forms are used by the printer (the tractor feed attachment must be present).

***CUT:** Single-cut sheets are used by the printer. Each sheet must be manually loaded. For cut sheets, the forms alignment message is not issued.

***AUTOCUT:** The sheet-feed attachment must be on the printer. Single-cut sheets are automatically fed into the printer. The forms alignment message is not sent for cut sheets.

MULTIUP

Specifies, for spooled output only, whether multiple pages of output are printed on a single physical page. This parameter is used only when the printer device type is *SCS or *IPDS.

***SAME:** The value does not change.

1: One page of output is printed on one physical sheet of paper.

2: Two pages of output are printed on 1 physical sheet of paper.

4: Four pages of output are printed on 1 physical sheet of paper.

DUPLEX

Specifies whether output is printed on one side or two sides of the paper.

***SAME:** The value does not change.

***NO:** The output is printed on one side of the paper.

***YES:** The output is printed on both sides of the paper with the top of each printed page at the same end of the paper.

***TUMBLE:** The output is printed on both sides of the paper with the top of one side at the opposite end of the top of the other side. This is usually used for output that will be bound at the top.

***FORMDF:** The duplex value specified in the form definition file is used. This value is valid only with printer device types of *AFPDS, *AFPDSLINE, or *LINE.

FRONTOVL

Specifies the qualified name of the object that contains both the overlay that is printed on the front side of the page and the offset, down and across, from the point of origin used when the overlay is printed. This parameter is used only when the printer device type is *SCS or *IPDS.

***SAME:** The value does not change.

***NONE:** No overlay is used.

The name of the overlay can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

Element 1: Overlay Name

overlay: Specify the name of the overlay.

Element 2: Offset Down

***SAME:** The value does not change.

offset-down: Specify the offset down from the point of origin at which to begin printing the overlay. If UOM(*CM) was specified on the CRTPRTF command when this file was created, valid values range from 0 through 57.79, and if UOM(*INCHES) was specified, valid values range from 0 through 22.57.

Element 3: Offset Across

***SAME:** The value does not change.

offset-across: Specify the offset across from the point of origin at which to begin printing the overlay. If UOM(*CM) was specified on the CRTPRTF command when this file was created, valid values range from 0 through 57.79, and if UOM(*INCHES) was specified, valid values range from 0 through 22.57.

BACKOVL

Specifies the qualified name of the object that contains both the overlay that is printed on the BACK side of the page and the offset, down and across, from the point of origin used when the overlay is printed. This parameter is used only when the printer device type is *SCS or *IPDS.

***SAME:** The value does not change.

***FRONTOVL:** The values specified on the FRONTOVL parameter are used.

***NONE:** No overlay is used.

The name of the overlay can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

Element 1: Overlay Name

overlay: Specify the name of the overlay.

Element 2: Offset Down

***SAME:** The value does not change.

offset-down: Specify the offset down from the point of origin at which to begin printing the overlay. If UOM(*CM) was specified on the CRTPRTF command when this file was created, valid values range from 0 through 57.79, and if UOM(*INCHES) was specified, valid values range from 0 through 22.57.

Element 3: Offset Across

***SAME:** The value does not change.

offset-across: Specify the offset across from the point of origin at which to begin printing the overlay. If UOM(*CM) was specified on the CRTPRTF command when this file was created, valid values range from 0 through 57.79, and if UOM(*INCHES) was specified, valid values range from 0 through 22.57.

FIDELITY

Specifies the print fidelity that is maintained for this file when printing to an AFP enabled printer.

***SAME:** The value does not change.

***ABSOLUTE:** Print the file exactly as intended. Stop printing the file immediately if an error is encountered in the data stream.

***CONTENT:** Try to override any errors in the data stream and attempt to continue printing the file.

VOL

Specifies, for diskette output files only, one or more volume identifiers of the diskettes on which this spooled file is written. The diskettes (volumes) must be placed in the device in the same order as the identifiers are specified in this parameter; a message is sent to the system operator if the order is different. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***NONE:** No diskette volume identifiers are specified. This output file is written on the first available diskette, based on the diskette writer's current position. No volume identifier checking is performed.

volume-identifier: Specify the identifiers of one or more volumes in the order in which they are inserted and used for this output file. No more identifiers can be specified than were initially specified for the diskette device file.

Each volume identifier contains up to six characters. A blank is used as the separator character when listing multiple identifiers. The number of volumes possible in the list is 50, but if more than 10 volume names are specified when the file is first opened, then only that number of files can be entered on the change command. Up to 10 volumes can be specified.

LABEL

Specifies, for diskette output files only, the data file identifier of the data file written on diskette from this spooled file. The data file identifier is stored in a label in the volume label area of the diskette.

***SAME:** The value does not change.

data-file-identifier: Specify the identifier (up to 8 characters) assigned to the data file being written on diskette from this spooled file.

EXCHTYPE

Specifies the exchange type used to write the spooled file. This parameter must be coded EXCHTYPE(*SAME) if the spooled file is not a diskette file.

***SAME:** The value does not change.

***STD:** The basic exchange format is used for a type 1 or a type 2 diskette. The H exchange type is used for a type 2D diskette.

***BASIC:** The basic exchange type is used.

***H:** The H exchange type is used.

***I:** The I exchange type is used.

CODE

Specifies the character code used. The code can be either extended binary-coded decimal interchange code (*EBCDIC) or the American National Standard Code for Information Interchange (*ASCII).

***SAME:** The value does not change.

***EBCDIC:** The extended binary-coded decimal interchange code (EBCDIC) character set code is used.

***ASCII:** The ASCII character set code is used.

Examples

Example 1: Moving a File to Another Queue

```
CHGSPLFA  FILE(SALES)  JOB(000147/JONES/BILLING)
  OUTQ(QPRINT2)  FORMTYPE('1140-6')
```

This command moves the file named SALES (of the BILLING job numbered 000147) from the present queue to the QPRINT2 queue. It also changes the forms identifier to 1140-6, which means that this form type is used in the printer.

Example 2: Changing Number of Output Copies

```
CHGSPLFA  FILE(DEPT511)  COPIES(2)  FILESEP(5)
```

This command changes the attributes of the spooled file DEPT511 that is produced by the submitter's job. It changes the number of output copies to 2 and specifies that five separator pages precede each copy.

Example 3: Changing Starting and Ending Pages to Print

```
CHGSPLFA  FILE(DEPT481)  PAGERANGE(99 100)
```

This command changes the attributes of the spooled file, DEPT481. It changes the starting and ending pages that are to be printed. Now, only pages 99 and 100 of each copy of the file is printed.

Example 4: Starting on a Specific Page

```
CHGSPLFA  FILE(DEPT481)  RESTART(5)
```

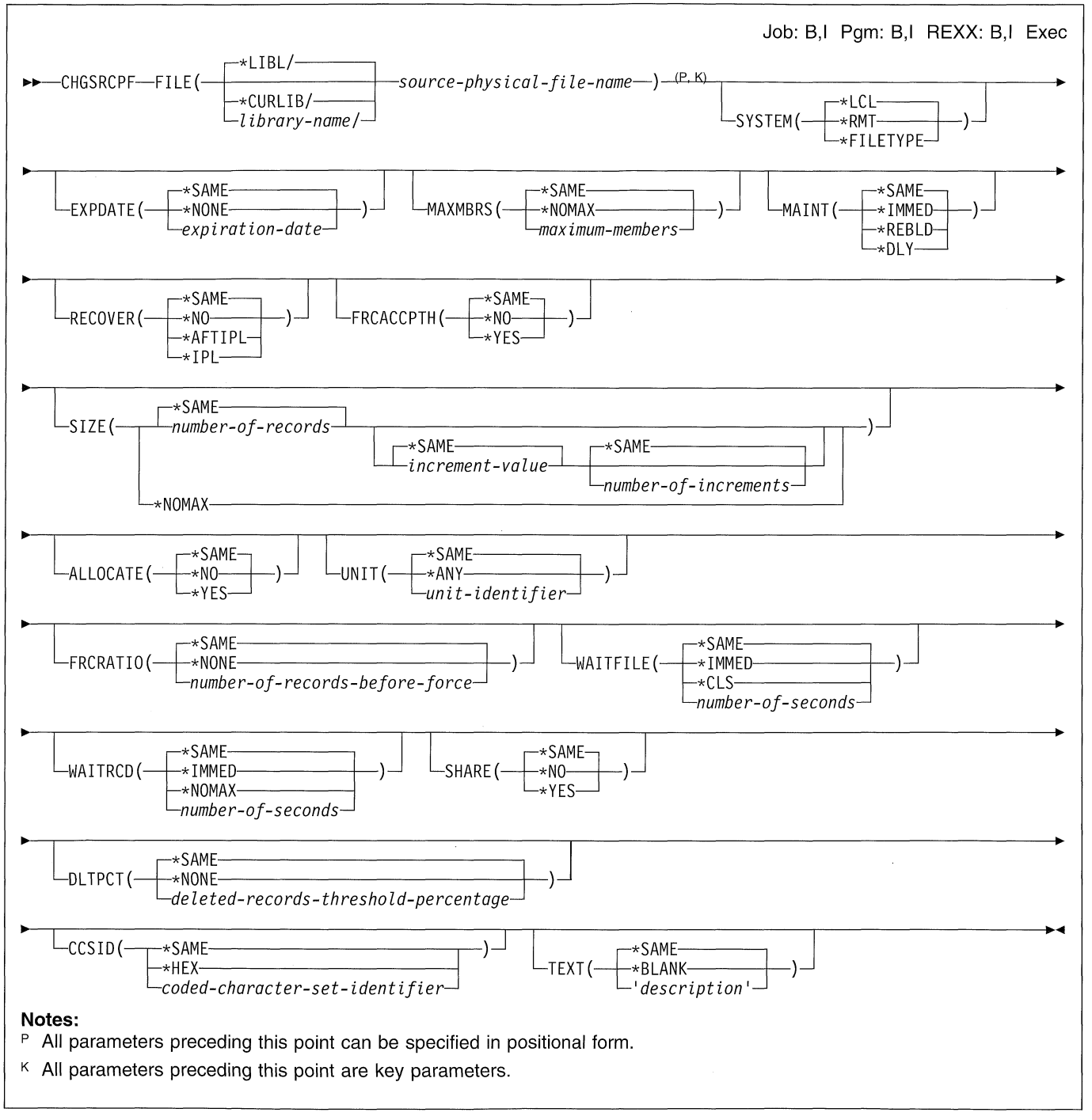
This command restarts printing spooled file DEPT481 on page 5. All of the copies that follow are printed from the specified starting page to ending page. If the file is in WTR status, the writer stops printing the current copy and restarts printing on page 5. The page specified on the RESTART parameter must be within the range specified on the PAGERANGE parameter.

Example 5: Restarting on the Next Page

```
CHGSPLFA  FILE(DEPT481)  RESTART(*NEXT)
```

This command restarts the printing job on the page following the last page printed when the job was interrupted. All of the copies that follow are printed from the specified starting page to ending page. The file must not be in WTR status. If the file is in WTR status, this command is rejected and a message is sent to the user. RESTART(*NEXT) is not valid when a file is being processed by a writer.

CHGSRCPF (Change Source Physical File) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Source Physical File (CHGSRCPF) command changes the attributes of a source physical file and its members. The changed attributes are also used for all members subsequently added to the file.

Restrictions:

1. To change a source physical file, the user must have object management authority and object operational authority for the file and read authority to the library.
2. For the user to change the file, an exclusive lock is necessary; no one may be using the file for any purpose.

Required Parameter

FILE

Specifies the qualified name of the source physical file that is changed.

The name of the source physical file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

source-physical-file-name: Specify the name of the source physical file being changed.

Note: If a DDM file is specified, the source physical file specified on the RMTFILE parameter of the Create Distributed Data Management File (CRTDDMF) command is changed on the remote system specified on the RMTLOCNAME parameter on the CRTDDMF command. More information is outlined in the SYSTEM parameter of this command.

Optional Parameters

SYSTEM

Specifies whether the source physical file is changed on the local system or the remote system.

***LCL:** The source physical file is changed on the local system.

***RMT:** The source physical file is changed on a remote system using DDM. The source physical file name specified on the FILE parameter must be the name of the DDM file (created by using the Create Distributed Data Management File (CRTDDMF) command). The DDM file contains the name of the source physical file to be changed (RMTFILE parameter on the CRTDDMF command) and the name of the remote system (RMTLOCNAME parameter on the CRTDDMF command) on which the file is to be changed.

***FILETYPE:** If the name specified on the FILE parameter is a DDM file, the source physical file is changed on the remote system specified by the RMTLOCNAME parameter of that DDM file. Otherwise, the name specified on the FILE parameter to be changed must be the name of a local source physical file.

EXPDATE

Specifies the expiration date. The files cannot be overwritten until the expiration date. The expiration date must be later than or equal to the current date.

Note: An attempt to open a file member that has exceeded its expiration date causes an error message to be sent. The date must be specified in the format defined by the job attributes, QDATFMT and QDATSEP, and must be enclosed in apostrophes, if special characters are used in the format.

***SAME:** The expiration date of the file does not change.

***NONE:** No expiration date is specified.

expiration-date: Specify the date after which the member is not used.

MAXMBRS

Specifies the maximum number of members that the physical file can have at any time. The maximum number of members specified must be greater than or equal to the current number of members in the file.

***SAME:** The value does not change.

***NOMAX:** The system maximum is used.

maximum-members: Specify the value for the maximum number of members that the physical file can have. Valid values range from 1 through 32767.

MAINT

Specifies the type of access path maintenance used for all members of the physical file. This parameter is valid only if a keyed access path is used.

***SAME:** The value does not change.

***IMMED:** The access path is maintained for each physical file member whether the source physical file is opened or closed. The access path is changed whenever a record is updated, added to, or deleted from a member of this file or a logical file member based on a member of this file.

***REBLD:** The access path is rebuilt when a file member is opened during the running of the program. The access path is continuously maintained until the member is closed; then the access path maintenance is ended.

***DLY:** The maintenance of the access path is delayed until the member is opened for use. Then the access path is changed only for records that are added, deleted, or changed since the file was last closed. (While the file is open, all changes made to based-on members are immediately reflected in the access paths of the members of the opened files, no matter what is specified for MAINT.) To prevent a lengthy rebuild time when the file is opened, *DLY should be specified only when the number of changes to the access path between a close and the next open are small (when key fields in records for this access path change infrequently).

If the number of changes saved reaches approximately 10% of the access path size, the system stops saving changes and the access path is completely rebuilt the next time the file is opened.

RECOVER

Specifies, for files having immediate or delayed maintenance on their access paths, when recovery processing

of the file is done if a system failure occurs while the access path is being changed.

The access path having immediate or delayed maintenance is rebuilt during initial program load (IPL) (before any user can run a job), after the IPL is completed (while other jobs are running), or the next time the file is opened. While the access path is being rebuilt, the file cannot be used by any job.

During the IPL, an Override Access Path Recovery display will list those access paths that must be recovered and what the RECOVER parameter value is for each. The user can override the RECOVER parameter value on this display. More information is in the *Advanced Backup and Recovery Guide*.

The access path having rebuild maintenance is rebuilt the next time its file is opened, the time that it normally is rebuilt. This parameter is valid only for files with a keyed access path.

***SAME:** The value does not change.

***NO:** The access path of the file is not rebuilt. The file's access path, if not valid, is rebuilt the next time the file is opened.

***AFTIPL:** The access path of the file is rebuilt after the IPL operation is completed. This option allows other jobs not using this file to start processing immediately after the IPL has been completed. If a job tries to open the file while its access path is being rebuilt, a file open exception occurs.

***IPL:** The access path of the file is rebuilt during the IPL. This ensures that the file's access path is rebuilt before the first user program tries to use it; however, no jobs can start running until after all files that specify RECOVER(*IPL) have their access paths rebuilt.

FRCACCPTH

Specifies, for files with keyed access paths only, whether access path changes are forced to auxiliary storage along with the associated records in the file whenever the access path is changed. FRCACCPTH(*YES) minimizes (but does not remove) the possibility that an abnormal job end can cause damage to the access path, which then requires it to be rebuilt.

***SAME:** The value does not change.

***NO:** The access path and changed records are not forced to auxiliary storage whenever the access path changes.

***YES:** The access path and changed records are forced to auxiliary storage whenever the access path is changed. If this value is specified, MAINT(*REBLD) cannot be specified.

FRCACCPTH(*YES) slows the response time of the system if the access path is changed in an interactive job. If the access path is changed frequently, the entire performance of the system is affected.

SIZE

Specifies the initial number of records in each member of the file, the number of records in each increment that is automatically added to the member size, and the number of times the increment is automatically applied. The change in the initial number of records takes effect when a new member is added to the file or when a current member is cleared, restored, or reorganized.

A change to the number of records to add for each increment, and the number of times the increment is automatically applied, takes effect the next time a member of the file needs an increment.

The total size of the member (first number of records plus number of records added per increment times the number of increments) must be larger than the current size of the member. If it is smaller, an error message is issued, and the size does not change.

Element 1: Number of Records

***SAME:** The value does not change.

number-of-records: Specify the number of records (ranging from 1 through 16777215) that can be inserted before an automatic extension occurs. If automatic extensions are not wanted, enter zeros for the second and third values in the list.

Element 2: Increment Value

***SAME:** The value does not change.

increment-value: Specify the number of additional records (ranging from 0 through 32767) which, if greater than 10% of the size of the member when the maximum number of records is reached, are to be added to the member during an automatic extension.

If the number specified is not greater than 10% of the member size and not equal to zero, the member size is increased by 10%.

Specify 0 to prevent automatic extensions. This value must be 0 if the value for the number of increments is 0.

Element 3: Maximum Number of Increments

***SAME:** The value does not change.

number-of-increments: Specify the maximum number of increments (ranging from 0 through 32767) that can be automatically added to the member. Enter 0 to prevent automatic extensions. If the increment value is 0, the number of increments must be 0.

Other Single Values

***NOMAX:** The system maximum is used. This option cannot be specified if ALLOCATE(*YES) is in effect.

ALLOCATE

Specifies whether storage space is allocated for the initial number of records (SIZE parameter) for each physical file member when it is added. This change takes effect the next time a new member is added to the file or when a current member is cleared, restored, or reorganized.

***SAME:** The value does not change.

***NO:** When a new member is added, or when an existing member is cleared, restored, or reorganized, the system determines whether additional space is needed and allocates that amount.

***YES:** The amount of storage space specified in the first value of the SIZE parameter is allocated each time a new member is added or an existing member is cleared, restored, or reorganized. If that amount of storage space is not available, the member is not added, and a message is sent to the user. If this parameter value is used, SIZE(*NOMAX) must not be in effect.

UNIT

Specifies that the user can change the unit identifier of the auxiliary storage unit where the system attempts to allocate storage space for the file and its members as well as their associated access paths. This change takes effect for any increments that are applied to existing members or for new members to the file.

The system attempts to put new increments for a member on the specified unit. However, the space allocated before the change in the unit specification remains on the unit on which it was originally allocated. To have the system attempt to place an entire member on the specified unit, (1) save the member, (2) delete the member from the system, and (3) restore the member on the system.

The unit identifier is a number ranging from 1 through 255, assigned when a new disk device is configured. The user can display and change the configured disk devices using the Work with Disk Devices display from the Start System Services Tool (STRSST) command. More information on SST is in the *Advanced Backup and Recovery Guide*.

***SAME:** The value does not change.

***ANY:** The storage space for the file and its members is allocated on any available auxiliary storage unit.

unit-identifier: Specifies the storage unit where the system attempts to allocate the storage space for the file and for all its members.

FRCRATIO

The force write ratio parameter specifies the number of inserted, updated, or deleted records that are processed before they are forced to auxiliary (permanent) storage. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

If the physical file is being journaled, a larger force write ratio or *NONE should be specified. More information on Journal Management is in the *Advanced Backup and Recovery Guide*.

***SAME:** The value does not change.

***NONE:** There is no force write ratio; the system determines when the records are written to auxiliary storage.

number-of-records-before-force: Specify the number of inserted, updated, or deleted records that are processed before they are explicitly forced to auxiliary storage.

WAITFILE

Specifies the number of seconds that the program waits for the file resources and session resources to be allocated when the file is opened, or for the device or session resources to be allocated when an acquire operation is performed to the file. If those resources are not allocated within the specified wait time, an error message is sent to the program. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

Note: An immediate allocation of the device by the device resource is required when an acquire operation is performed to the file.

***SAME:** The value does not change.

***IMMED:** The program does not wait; when the file is opened, an immediate allocation of the file resources is required.

***CLS:** The job default wait time is used as the wait time for the file resources being allocated.

number-of-seconds: Specify the number of seconds that a program waits for the file resources to be allocated to the job. Valid values range from 1 through 32767 seconds.

WAITRCD

Specifies the number of seconds that a program waits for a record to be updated or deleted, or for a record read in the commitment control environment with LCKLVL(*ALL) specified. More information on record locking is in the *Database Guide*. If the record is not allocated in the specified wait time, an error message is sent to the program.

***SAME:** The value does not change.

***IMMED:** The program does not wait; when a record is locked, an immediate allocation of the record is required.

***NOMAX:** The system maximum is used.

number-of-seconds: Specify the number of seconds that a program waits for the file resources to be allocated to the job. Valid values range from 1 through 32767 seconds.

SHARE

Specifies whether the open data path (ODP) for the source physical file is shared with other programs in the routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

More information on shared database files is in the *Database Guide*.

***SAME:** The value does not change.

***NO:** The ODP created by the program with this attribute is not shared with other programs in the routing

step. Every time a program opens the file with this attribute, a new ODP to the file is created and activated.

***YES:** The ODP created with this attribute is shared with each program in the routing step that also specifies SHARE(*YES) when it opens the file.

Note: When SHARE(*YES) is specified and control is passed to a program, a read operation in that program retrieves the next input record. A write operation produces the next output record.

DLTPCT

Specifies the maximum percentage of deleted records that any member in the physical file can have. The percentage is based on the number of deleted records compared with the total record count in a member. This change takes effect the next time the file is opened and closed.

***SAME:** The value does not change.

***NONE:** No percentage is specified; the number of deleted records in the file members is not checked when a member is closed.

deleted-records-threshold-percentage: Specify the largest percentage of deleted records (ranging from 1 through 100) that any member in the file can have. If a value is larger than this percentage, a message is sent to the system history log (QHST) when the file is closed.

CCSID

Specifies the coded character set identifier (CCSID) used to describe character data in the fields of the file.

Note: The CCSID cannot be changed if:

- There are any logical files defined over the source physical file.
- Any explicit field- or file-level CCSIDs are specified on the CCSID keyword in DDS, or on the CCSID parameter of the CRTSRCPF and CRTPF commands, for fields in the physical file.
- Another file shares the physical file's format.

***SAME:** The CCSID does not change.

***HEX:** The CCSID 65535 is used, which indicates that the character data in the fields is treated as bit data and is not converted.

coded-character-set-identifier: Specify the CCSID being used. If a DBCS field is in the source physical file, the CCSID specified must have a corresponding mixed CCSID. More information on valid CCSIDs is in the *National Language Support Planning Guide*.

TEXT

Specifies text that briefly describes the source physical file. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Examples

Example 1: Changing the Expiration Date

```
CHGSRCPF   FILE(QGPL/INV)  EXPDATE('10/31/88')
```

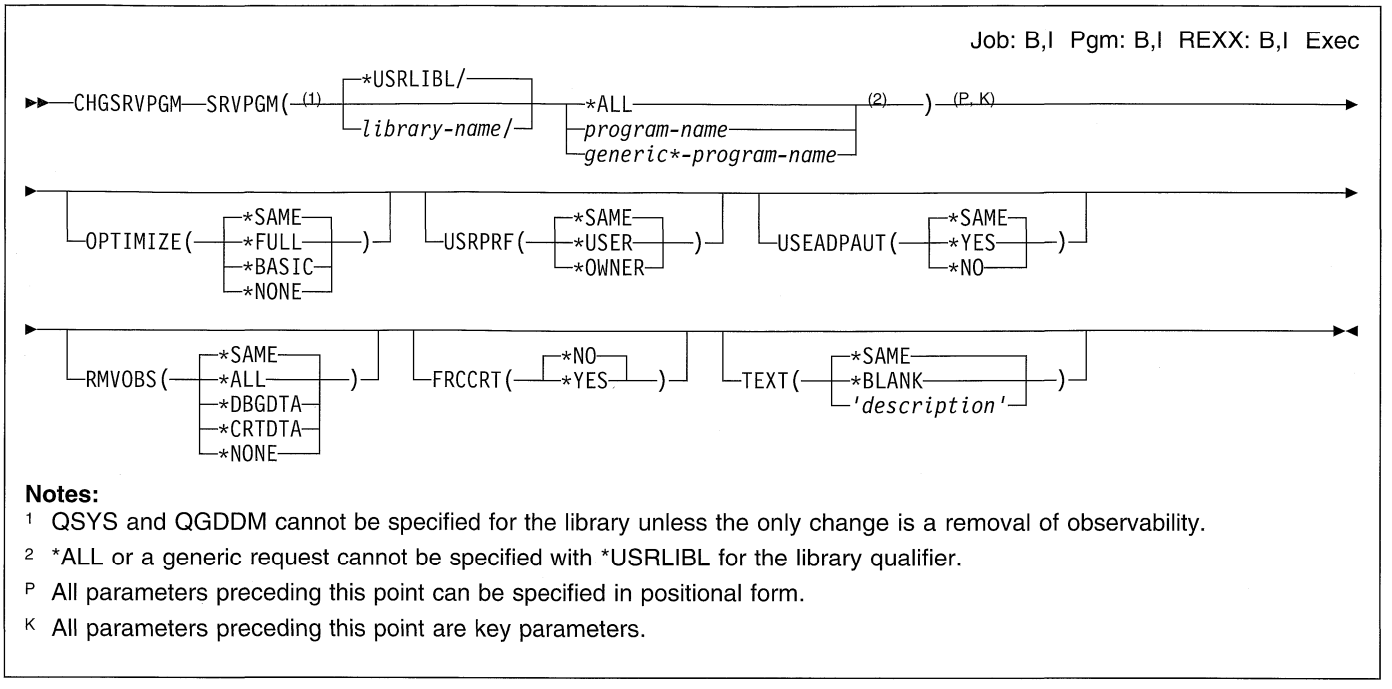
This command changes the expiration date of all members in file INV to October 31, 1988.

Example 2: Changing Text

```
CHGSRCF  FILE(QGPL/DDMF)  TEXT('Inventory File')
  SYSTEM(*RMT)
```

This command changes the text of file INV located in the QGPL library on the remote system. Prior to specifying the above command, this user had created a DDM file by specifying the command, CRTDDMF FILE(QGPL/DDMF) RMTFILE(QGPL/INV) RMTLOCNAME(AS400).

CHGSRVPGM (Change Service Program) Command



Notes:

1 QSYS and QGDDM cannot be specified for the library unless the only change is a removal of observability.

2 *ALL or a generic request cannot be specified with *USRLIBL for the library qualifier.

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change Service Program (CHGSRVPGM) command changes the attributes of a program without requiring that it be recompiled. The attributes that can be changed are the optimization attribute, the user profile attribute, the use-adopted-authority attribute, and the service program text. The user can also force re-creation of a service program even if the attributes being specified are the same as the current attributes.

Restrictions:

1. The user must have *USE authority for the library for the service program that is being changed.
2. The user must have *USE and *OBJMGT authority for the service program that is being changed.
3. The user must have *USE, *DLT, and *ADD authority for the library to change the optimization attribute (OPTIMIZE) or to force service program re-creation by specifying FRCCRT(*YES).
4. Only the service program owner, or a user with *SECADM and *ALLOBJ authority, can change the user profile attribute (USRPRF) or the use adopted authority attribute (USEADPAUT).
5. Service programs in library QSYS and QGDDM cannot be changed unless the only indicated change is a removal of observability.

Jobs running the service program do not fail.

Required Parameters

SRVPGM

Specifies the service programs whose attributes are being changed. *USRLIBL cannot be specified or defaulted for the library qualifier when a generic name or *ALL is specified for the program qualifier.

The name of the service program can be qualified by one of the following library values:

***USRLIBL:** Only the libraries in the user portion of the job's library list are searched.

library-name: Specify the name of the library to be searched.

***ALL:** All service programs in the specified library to which the user has some authority (for example, *USE authority) are selected for change.

program-name: Specify the name of the service programs whose attributes are being changed.

generic-program-name:* Specify the generic name of the service program. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be changed only if *ALL or *ALLUSR library values can be specified for the name. For more infor-

mation on the use of generic functions, refer to "Rules for Specifying Names."

Optional Parameters

OPTIMIZE
Specifies whether the service program is optimized. This parameter removes redundant instructions from the specified programs. Changing the current optimization level of a service program causes the system to re-create the service program with the new optimization level.

***SAME:** The value does not change.

***FULL:** The service program is fully optimized. In most cases, optimized service programs make more efficient use of system resources. When debugging ILE service programs at this level, variables can be displayed, but the displayed value may not be the current value. Variables can also be changed but using the variables changed at this level may cause unexpected results.

***BASIC:** Some optimization is performed on ILE service programs. When debugging ILE service programs at this level, variables can be displayed, but the displayed value may not be the current value. Variables can also be changed but using the variables changed at this level may cause unexpected results.

***NONE:** The program is not optimized. Variables can be displayed and changed when debugging ILE programs at this optimization level.

USRPRF
Specifies whether the authority checking done while this service program is running includes only the user who is running the service program (*USER) or both the user running the service program and the service program owner (*OWNER). The profiles of the service program user or both the service program user and the service program owner are used to control which objects can be used by the service program, including the authority the service program has for each object. Only the service program owner or a user with QSECOFR authority can change the user profile attribute.

***SAME:** The value does not change.

***USER:** The service program runs under the user profile of the service program's user.

***OWNER:** The user profiles of both the service program's owner and the service program's user are used when the service program is processed. The collective sets of object authority in both user profiles are used to find and access objects during service program processing. Authority from the owning user profile's group profile is not included in the authority for the running service program.

USEADPAUT
Specifies whether service program adopted authority from previous programs or service programs in the call stack are used as a source of authority when this service program is running.

***SAME:** The value does not change.

***YES:** Program or service program adopted authority from previous recursion levels is used when this service program is running.

***NO:** Program or service program adopted authority from previous recursion levels is not used when this service program is running.

RMVOBS
Specifies whether the observable information associated with service programs is removed.

***SAME:** The value does not change.

***ALL:** All of the observable information associated with the service program is removed, if possible. If the service program requires the observable information to ensure that it runs correctly, that information is not removed.

***DBGDTA:** All of the observable information necessary to allow the service program to be debugged is removed.

***CRTDTA:** All of the observable information necessary to allow the service program to be re-created or to change the optimization level is removed.

***NONE:** None of the observable information associated with the service program is removed.

FRCCRT
Specifies whether service program re-creation is forced.

***NO:** Service program re-creation is not forced unless the optimization level (OPTIMIZE), use adopted authority (USEADPAUT), or user profile (USRPRF) parameters are changed. This option allows the system to determine whether a change is required.

***YES:** Service program re-creation is forced whether or not the optimization level (OPTIMIZE), use adopted authority (USEADPAUT) or user profile (USRPRF) parameter has been changed.

TEXT
Specifies text that briefly describes the service program. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Examples

Example 1: Optimizing a Service Program

```
CHGSRVPGM   SRVPGM(PROG1/SERVICE)  OPTIMIZE(*FULL)
  USRPRF(*OWNER)
```

The service program SERVICE in library PROG1 is optimized, and the user profile under which it is processed is changed to include the service program owner's user profile. Only the owner of service program PROG1/SERVICE, or a user with security officer authority, can change the USRPRF attribute. The service program is re-created only if the attributes specified differ from those of the current service program.

Example 2: Changing Text for a Service Program

```
CHGSRVPGM    PGM(*USRLIBL/KNUTE)
  TEXT('Service program description')
```

This command changes the text for service program KNUTE. The user portion of the library list is used to find the service program.

Example 3: Optimizing Multiple Service Programs

```
CHGSRVPGM    SRVPGM(PROG1/ACE*)  OPTIMIZE(*FULL)
```

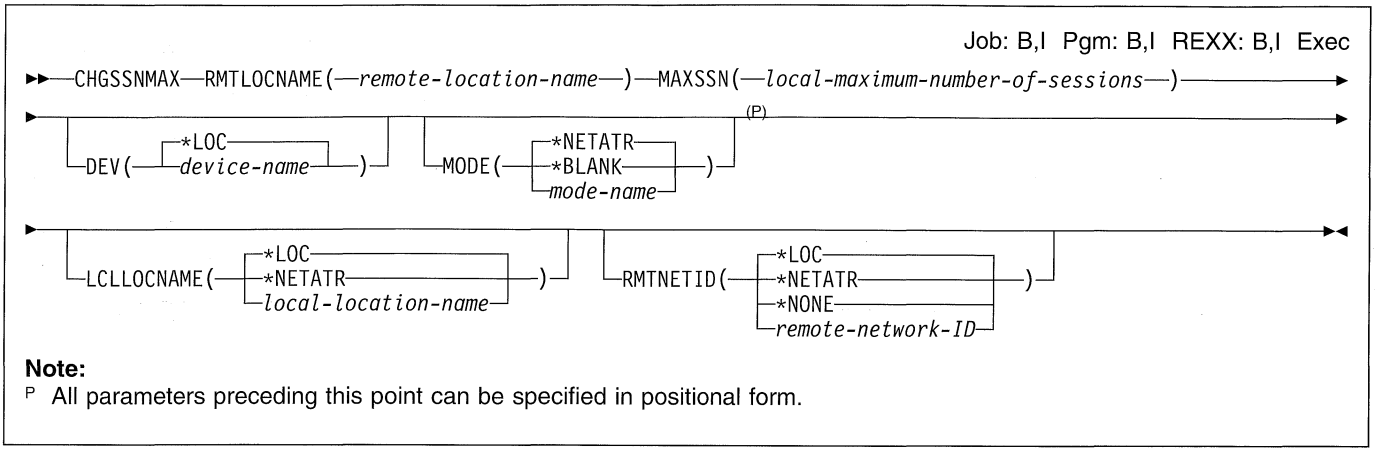
All service programs in library PROG1 that have not been optimized, and whose names begin with ACE, are optimized.

Example 4: Changing Text of Multiple Service Programs

```
CHGSRVPGM    SRVPGM(PROG2/*ALL)  TEXT('Generic Text')
```

This command changes the text of all service programs in library PROG2 to Generic Text.

CHGSSNMAX (Change Session Maximum) Command



Note:

P All parameters preceding this point can be specified in positional form.

Purpose

The Change Session Maximum (CHGSSNMAX) command is used to dynamically change the maximum number of sessions the local location allows to an advanced program-to-program communications (APPC) mode. When a change to the MAXSSN parameter is made, the remote system is informed and allowed to negotiate for a lower session maximum. The remote system cannot negotiate a session maximum higher than the value specified for the local number of sessions specified. The resulting session maximum value is the *current session maximum.* Neither system may activate more sessions than the current session maximum. If the requested session maximum is accepted or negotiated by the remote system, the value requested on the CHGSSNMAX command is stored as the *local session maximum*; the remote system is not allowed to increase the current session maximum above the value stored as the local session maximum. This new value for the local session is only used the next time a new session maximum needs to be negotiated. The current session maximum, which controls how many sessions can be active between the local and remote location, is not changed if the command fails.

If the request to change the session maximum is rejected by the remote system, the CHGSSNMAX command ends abnormally and the local session maximum is changed as follows: If it is increasing, it is changed to the value specified; if it is decreasing, it is not changed.

This command is normally used by the system operator to control the number of sessions that can be active at the same time with a remote system. This command is used only when the remote location is active and the mode is in a *started* state. If the current number of active sessions is greater than the maximum number specified on the command, no new sessions are created until the number of active sessions falls below that specified on the command. If the current number of active sessions is less than the maximum number specified, sessions are not created until jobs requiring them are started.

The value created by the systems remains in effect until another CHGSSNMAX command or an End Mode (ENDMOD) command is run for the same mode, or until all the devices for the local location are varied off.

Many CHGSSNMAX commands can be sent before the maximum number of sessions is made active. The number specified the last time the command was called is the *local session maximum* value. If the device is varying off, this command ends with an error.

Notes:

1. When this command is used to reduce the number of sessions with a remote system, the sessions that are ended first are the available locally controlled sessions, followed by any other available sessions. If the new session count is still not reached, other sessions are ended as jobs using them are completed or are canceled.
2. When the CHGSSNMAX command is used to increase the maximum number of sessions that can be created with a remote system, the locally controlled sessions are made available first (depending on the negotiated values), and then other sessions are made available.
3. The CHGSSNMAX command does not change the value specified for the MAXSSN parameter in the mode description; the Change Mode Description (CHGMODD) command must be used to permanently change the value. When the device is next varied on, the MAXSSN value from the mode description is used to limit the number of sessions instead of the MAXSSN value specified on a previous CHGSSNMAX command.

The *APPC Programmer's Guide* has more information on this command.

Required Parameters

RMTLOCNAME
Specifies the name of the remote location that is used with this object.

MAXSSN
Specifies the number of sessions allowed with the remote system. This value represents the desired maximum session number for the specified mode name. It must be less than or equal to the MAXSSN parameter limit defined in the mode description. This value can be negotiated to a lower value by the remote system; therefore, the value specified here is not necessarily the value that is used. Valid values for this parameter are 1 through 512.

Optional Parameters

DEV
Specifies the name of the device description used with the remote location.

***LOC:** The device associated with the remote location is used. If several devices are associated with the remote location, the system determines which device is used.

device-name: Specify the name of a communications device associated with the remote location. If the device name is not valid for the remote location, a message is sent when the program device entry is acquired. More information on device names is in the *APPC Programmer's Guide*.

MODE
Specifies the name of the mode that is changed. The mode name cannot be SNASVCMG or CPSVCMG; these mode names are reserved for system use.

***NETATR:** The mode name specified in the network attributes is used.

***BLANK:** The mode name consisting of 8 blank characters is used.

mode-name: Specify a value, no more than 8 characters, used to identify the mode that is changed.

LCLLOCNAME
Specifies the local location name.

***LOC:** The device associated with the remote location is used. If several devices are associated with the remote location, the system determines which device is used.

***NETATR:** The LCLLOCNAME value specified in the system network attributes is used.

local-location-name: Specify the local location name associated with the remote location.

RMTNETID
Specifies the remote network ID that is used with the remote location.

***LOC:** The remote network identifier (ID) associated with the remote location is used. If several remote network IDs are associated with the remote location, the system determines which remote network ID is used.

***NETATR:** The RMTNETID value specified in the system network attributes is used.

***NONE:** No remote network identifier (ID) is used.

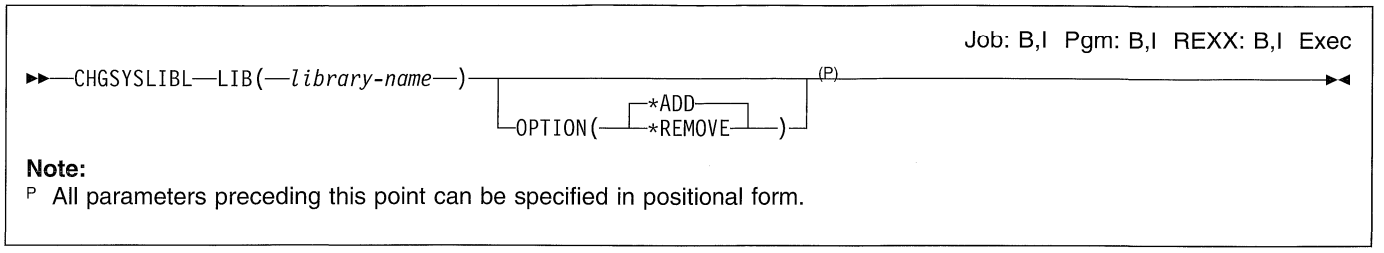
remote-network-ID: Specify a remote network ID for the remote location.

Example

```
CHGSSNMAX  RMTLOCNAME(APPCRLOC)  DEV(APPCDEV)
  MODE(APPC2)  MAXSSN(3)
```

This command changes the maximum number of sessions allowed by remote location APPCRLOC for mode APPC2 to a maximum of three.

CHGSYSLIBL (Change System Library List) Command



Note:
P All parameters preceding this point can be specified in positional form.

Purpose

The Change System Library List (CHGSYSLIBL) command changes the system portion of the library list for the job in which the command is entered. The user can specify if the library is added to the beginning of the system portion of the library list, or removed from that list.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. The library QSYS cannot be added to or removed from the system portion of the library list.
3. The CHGSYSLIBL command is authorized only to the security officer or a user with the *ALLOBJ special authority. The QSYSLIBL system value cannot be changed without such authority. However, the security officer can authorize individual users to the command, or can provide programs that adopt the security officer profile and can add a library on the system portion of the library list.

Required Parameter

LIB

Specifies the name of the library that either is added to, or is removed from, the system portion of the job's library list.

Optional Parameter

OPTION

Specifies whether the library is added to, or removed from, the system portion of the library list.

***ADD:** The specified library is added to the beginning of the system portion of the library list.

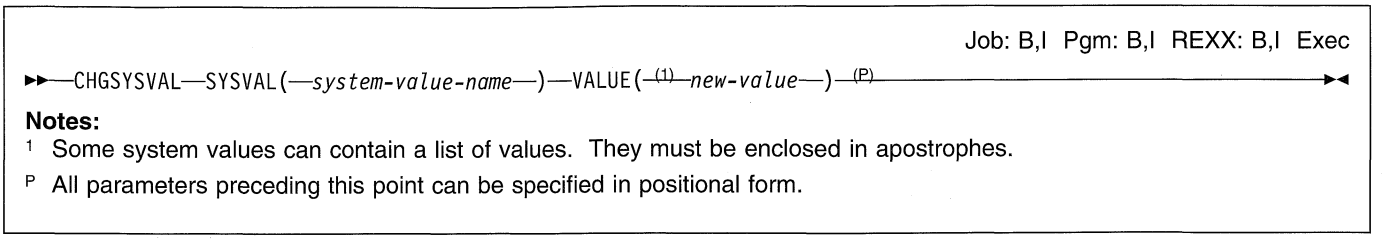
***REMOVE:** The specified library is removed from the system portion of the library list.

Example

```
CHGSYSLIBL  LIB(PAYROLL)  OPTION(*ADD)
```

This command adds the library PAYROLL to the beginning of the system portion of the library list.

CHGSYSVAL (Change System Value) Command



Purpose

The Change System Value (CHGSYSVAL) command changes the current value of the specified system value. System values are provided as part of the system. They are used by the system to control certain operations in the operating system and to communicate the status of certain conditions to the user. Changes to some system values take effect immediately, some do not take effect until new jobs are started, and others do not take effect until the next IPL. Note that if a change is made to a date or time system value during any operation that measures the length of time, a negative value may be set if the end time is less than the start time. More information about system values is in the *Work Management Guide*.

Restrictions:

1. This command is shipped with public *EXCLUDE authority and the QPGMR, QSYSOPR, and QSRV user profiles have private authorities to use the command.
2. You must have *ALLOBJ and *SECADM special authorities to change security related system values.

Required Parameters

SYSVAL

Specifies the name of the system value whose value is being changed. Most of the system values can be specified; however, some cannot have their values changed by this command. More information on which values can be specified is in the *Work Management Guide*.

VALUE

Specifies the new value of the system value. Some system values, such as QUSRLIBL and QCTLSBSD, may be made up of multiple character strings. These strings must be separated by blanks; apostrophes must surround the entire contents of the VALUE parameter.

For system values that accept alphabetic characters, with the exception of QPRTTXT, letters that are entered in lowercase (a–z) are translated into uppercase (A–Z) even if they are enclosed in apostrophes. Some system values, such as QDATE and QDBRCVYWT, are zoned-decimal values (character in nature) and must also be enclosed in apostrophes when specified in this parameter. For numeric system values, apostrophes cannot be used. Descriptions of all system values are in the *Work Management Guide*. Enter the new values that meet the type, length, and range requirements for that system value.

Examples

Example 1: Changing the System Value of a Character Variable

```
CHGSYSVAL   SYSVAL(QHOUR)  VALUE('12')
```

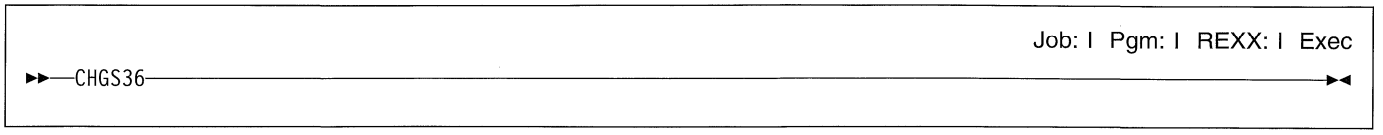
This command changes the value of the system value QHOUR (which is a subvalue of the QTIME system value) to 12. Because QHOUR is a character variable, 2 characters long, the system value is set to the character representation of 12, which is hex F1F2 and, therefore, must be enclosed in apostrophes. Also, the QTIME system value is updated with this value because QHOUR is a subvalue of QTIME.

Example 2: Changing a System Value to Contain Multiple Character Strings

```
CHGSYSVAL  SYSVAL(QUSRLIBL)
  VALUE('INVLIB STOCKLIB MYLIB')
```

This command changes the value of the system value QUSRLIBL, which specifies the default list of libraries in the user portion of the library list to be used for a job at the time the job is started. The user portion of the library list contains the libraries INVLIB, STOCKLIB, and MYLIB.

CHGS36 (Change System/36) Command



Purpose

The Change System/36 (CHGS36) command allows the user to change or update the description of the System/36 environment configuration. The original description of the System/36 environment is supplied by IBM.

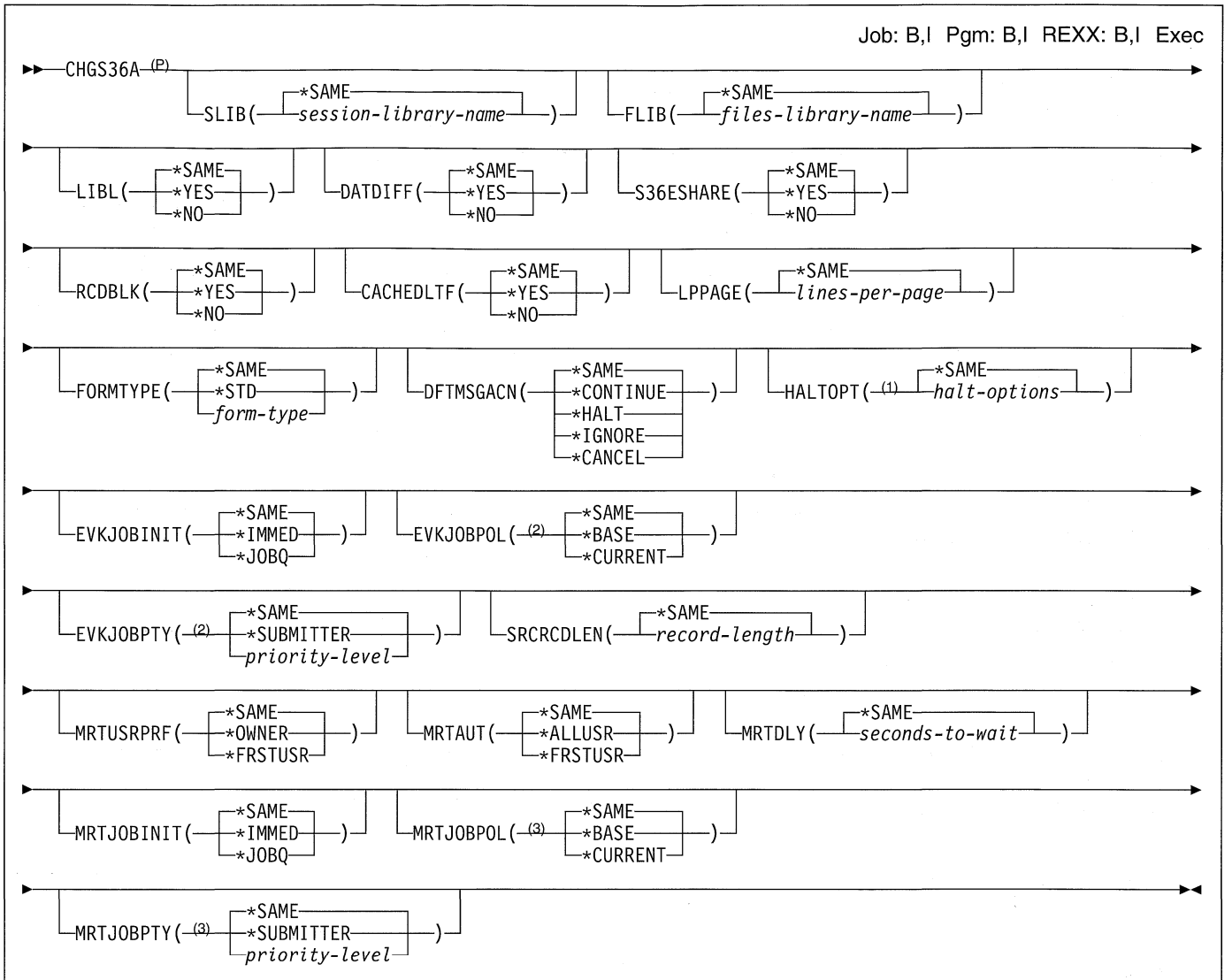
There are no parameters for this command.

Example

```
CHGS36
```

This command allows the user to change the System/36 Environment description. This command allows the user to change display stations, printers, tapes, diskettes, 3270 device emulation, general environment values, and, if authorized, MRT security values.

CHGS36A (Change System/36 Attributes) Command



Notes:

1 The HALTOPT parameter is valid only when DFTMSGACN(*HALT) is specified.

2 This parameter is valid only when EVKJOBINIT(*IMMED) is specified.

3 This parameter is valid only when MRTJOBINIT(*IMMED) is specified.

P All parameters preceding this point can be specified in positional form.

Purpose

The Change System/36 Attributes (CHGS36A) command allows the user to change the attributes of the System/36 environment configuration while the AS/400 system is running in the System/36 environment.

More information about the System/36 environment is in the *System Reference for the System/36 Environment*.

Optional Parameters

SLIB

Specifies the name of the default session library for users running jobs in the System/36 environment.

***SAME:** The value does not change.

session-library-name: Specify the name of the default session library.

FLIB

Specifies the default files library for users running jobs in the System/36 environment.

***SAME:** The value does not change.

files-library-name: Specify the name of the default files library.

LIBL

Specifies whether the library list is used when specifying database files for System/36 environment jobs.

***SAME:** The value does not change.

***YES:** The library list is used to search for database files.

***NO:** The library list is not used to search for database files.

DATDIFF

Specifies whether jobs running in the System/36 environment can use files of the same name, distinguished by creation date.

***SAME:** The value does not change.

***YES:** Jobs can use files of the same name if the files have different creation dates.

***NO:** Jobs cannot use files of the same name. Each file must have a unique name.

S36ESHARE

Specifies whether the System/36 environment opens database files in a way to allow an open data path (ODP) to be shared by multiple programs processing in the same job.

***SAME:** The value does not change.

***YES:** Programs share an ODP to database files opened during the job. The shared files are held open between job steps.

***NO:** Programs do not share an ODP to database files opened during the job. Files are closed between job steps.

RCDBLK

Specifies whether the System/36 environment uses record blocking for sequential database files that share an ODP. More information about record blocking is in the *Concepts and Programmer's Guide for the System/36 Environment*.

***SAME:** The value does not change.

***YES:** Jobs running use record blocking for shared sequential files.

***NO:** Jobs running do not use record blocking for shared sequential files.

CACHEDLTF

Specifies whether database files deleted by the System/36 environment are stored in a cache.

***SAME:** The value does not change.

***YES:** The deleted files are stored in a cache.

***NO:** The deleted files are not stored in a cache.

LPPAGE

Specifies the default number of lines printed on a page for all printers for jobs running in the System/36 environment when no lines-per-page are specified in the SET command or the FORMS or PRINTER OCL statement.

***SAME:** The value does not change.

lines-per-page: Specify the number of lines per page. Valid values range from 1 through 112.

FORMTYPE

Specifies the form type of the default printer form which is used for System/36 printouts when no form type is specified in the SET command or in the FORMS or PRINTER OCL statement. The form types used to indicate different printer forms are user-defined and can be a maximum of 4 characters in length.

***SAME:** The value does not change.

***STD:** The standard form type is used.

form-type: Specify a user-defined form type.

DFTMSGACN

Specifies the default action used for escape messages issued by CL commands in procedures running in the System/36 environment. The default action is used for messages not in the message list and when there is no message list. The default action is not used if the message list contains a message ID.

***SAME:** The value does not change.

***CONTINUE:** Processing continues with the next statement after the CL command. The ID of the escape message is saved and can be retrieved by the message ID substitution expression (?MSGID?).

***HALT:** The procedure is stopped. Processing continues as specified on the HALTOPT parameter.

***IGNORE:** The error is ignored and processing continues with the statement after the CL command. The ID of the escape message is not saved.

***CANCEL:** The procedure is canceled.

HALTOPT

Specifies a list of continuation options available when *HALT is specified for the default message action parameter. The list of options is a value consisting of up to 4 options with values ranging from 0 through 3, each of which represents an allowed response. If no options are specified, options 0 and 3 (value 03) are allowed. The meanings for the numbers assigned to the options are:

0 Continue. The message ID is saved for retrieval.
1 Retry the command. The message ID is not saved.
2 Cancel the job step. The message is saved for retrieval.
3 Cancel the job.

***SAME:** The value does not change.

halt-options: Specify up to four continuation options. Multiple options must be specified in ascending order and each option must be unique. Each digit must be the character 0, 1, 2, or 3 (blanks are ignored).

EVKJOBINIT
Specifies how EVOKE jobs or job steps are started within the System/36 environment.

***SAME:** The value does not change.

***IMMED:** The job queue is bypassed when starting jobs.

***JOBQ:** Jobs are started from the job queue.

EVKJOBPOL
Specifies the storage pool used for an EVOKE job that bypassed the job queue when starting to run in the System/36 environment.

***SAME:** The value does not change.

***BASE:** The job uses the subsystem's base pool storage area.

***CURRENT:** The job uses the same storage pool as the submitting job.

EVKJOBPTY
Specifies the priority level at which an EVOKE job in the System/36 environment must be started when it bypasses the job queue.

***SAME:** The value does not change.

***SUBMITTER:** The job is started with the same run priority as the submitting job.

priority-level: Specify a priority level. Valid values range from 1 through 99.

SRCRCDLEN
Specifies the record length in bytes for System/36 source files QS36PRC and QS36SRC. These source files are created by System/36 environment utilities.

***SAME:** The value does not change.

record-length: Specify the source file record length. Valid values range from 40 through 120 bytes (not including the extra 12 bytes required for the source sequence and date fields of each record).

MRTUSRPRF
Specifies the user profile under which the Multiple Requester Terminal (MRT) program runs to check security in the System/36 environment.

***SAME:** The value does not change.

***OWNER:** The MRT program runs under the profile of the MRT program owner.

***FRSTUSR:** The MRT program runs under the profile of the user that starts the program.

MRTAUT
Specifies which users are checked for their authority to obtain access to files used by the MRT program in the System/36 environment.

***SAME:** The value does not change.

***ALLUSR:** All users are checked for their authority to obtain access to the files.

***FRSTUSR:** Only users who start MRT programs are checked for their authority to obtain access to the files.

MRTDLY
Specifies the time (in seconds) that the system delays (waits) before ending the MRT program in the System/36 environment. The specified value is not valid if the program is a never-ending program (NEP).

***SAME:** The value does not change.

seconds-to-wait: Specify the number of seconds the system waits before ending the program. Valid values range from 0 to 32767 seconds.

MRTJOBINIT
Specifies how an MRT job is started in the System/36 environment.

***SAME:** The value does not change.

***IMMED:** The job queue is bypassed when starting the job.

***JOBQ:** The job is started from the job queue.

MRTJOBPOL
Specifies the storage pool used for an MRT job started without using the job queue in the System/36 environment.

***SAME:** The value does not change.

***BASE:** The job uses the subsystem's base pool storage area.

***CURRENT:** The job uses the same storage pool as the submitting job.

MRTJOBPTY
Specifies the priority level for starting an MRT job that bypasses the job queue.

***SAME:** The value does not change.

***SUBMITTER:** The job starts with the same priority level as the submitting job.

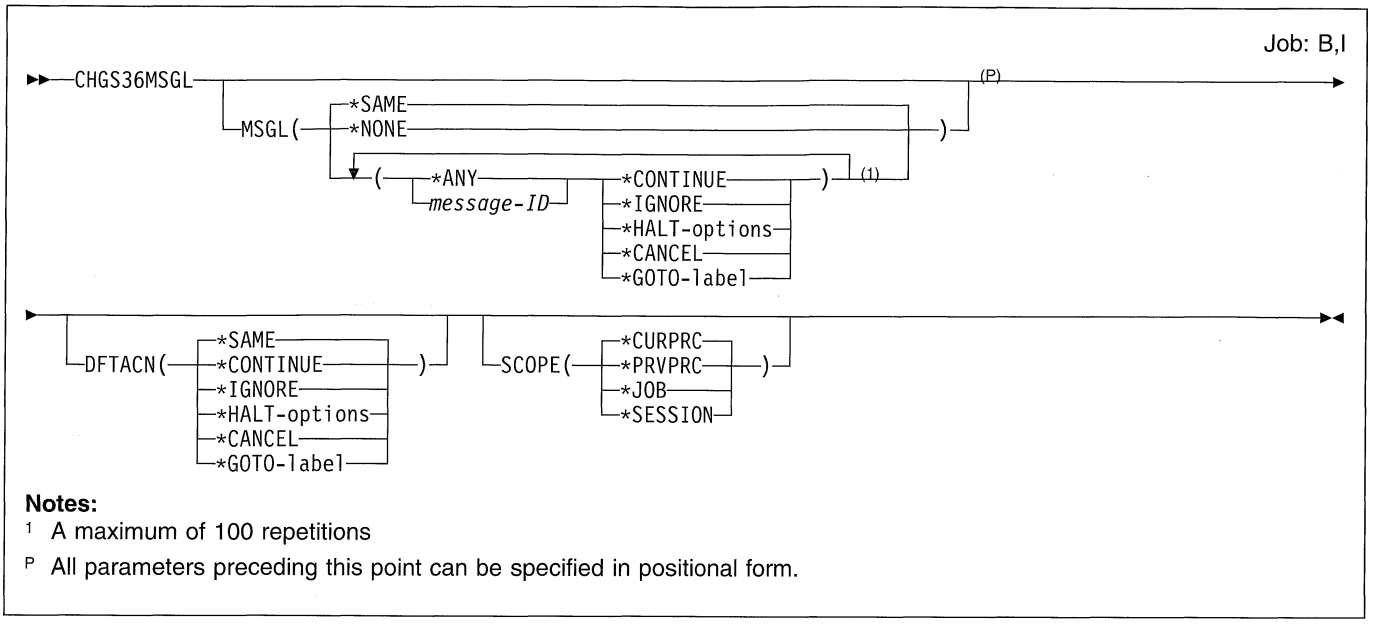
priority-level: Specify the priority level for starting the job. Valid values range from 1 through 99.

Example

```
CHGS36A   FLIB(MYLIB)  CACHEDLTF(*YES)  LPPAGE(66)
```

This command changes the value of the default files library to MYLIB for users running jobs in the System/36 environment. Storage for deleted files is changed to a cache. The number of lines printed on a page is changed to 66.

CHGS36MSGL (Change System/36 Message List) Command



Notes:
1 A maximum of 100 repetitions
P All parameters preceding this point can be specified in positional form.

Purpose

The Change System/36 Message List (CHGS36MSGL) command determines the action taken for specific escape messages issued by Control Language (CL) commands in a procedure running in the System/36 environment. It is also used to set the default action for escape messages that are not specified.

This command is allowed to be external to a procedure only when SCOPE(*JOB) or SCOPE(*SESSION) is specified. It is not allowed in CL programs. It is valid only when the System/36 Environment is active.

For System/36 environment jobs that are started by the JOBQ command, the // JOBQ OCL statement, or the // EVOKE OCL statement, the initial message default action is taken from the job level of the submitting job. For other jobs, it is taken from the System/36 environment configuration. The Change System/36 Configuration (CHGS36) command can be used to set the initial message default action in the configuration.

Optional Parameters

MSGL

Specifies a list of message IDs and the action that is taken for each message. One or more message IDs can be specified, along with the action that is taken for it. When a CL command issues an escape message, the message list is searched for the message ID and the specified action is taken. If the message ID is not found in the message list, the default action is taken.

The MSGL parameter can be specified only if SCOPE(*CURPRC) or SCOPE(*PRVPRC) is specified.

***SAME:** The value does not change.

***NONE:** The message list is removed.

Element 1: Message ID Added to List

***ANY:** The specified action is taken for any message ID not previously found in the message list. Because this value matches any message, it should be the last message ID specified in the list. Message IDs specified after this value are ignored.

message-ID: Specify the message ID being added to the list. Each message ID must be exactly 7 characters in length and must conform to the rules for message IDs. Generic message IDs can be specified by ending the message ID with either 00 or 0000. For example, CPF1200 would match all messages beginning with CPF12, and CPF0000 would match all messages beginning with CPF.

The message list is searched in the order that it is specified on the command. Therefore, if the message list contains more than one message ID that matches the message ID being searched for, the first one is used. For example, if the message list contains CPF1200 followed by CPF1234, and the message CPF1234 is being searched for, the generic message ID is found first, and the action specified for that message ID is taken.

Element 2: Action Taken for Message ID

***CONTINUE:** Processing continues with the next statement after the CL command. The ID of the escape message is saved and can be retrieved by the ?MSGID? substitution expression.

***IGNORE:** Processing continues with the next statement after the CL command. The ID of the escape

message is not saved, and the ?MSGID? substitution expression is null.

***HALT-options:** A halt with options is issued. This value may optionally be followed by a list of options allowed on the halt. If the options allowed are not specified, options 0 (continue) and 3 (cancel) are allowed.

The list of options is a value consisting of up to 4 numbers ranging from 0 through 3, each of which represents an option that is allowed in response to the halt. The meaning of each possible value in the list follows:

0 Continue. The ?MSGID? substitution expression is set.

1 Retry the command. The ?MSGID? substitution expression is not set.

2 Cancel the job step. The ?MSGID? substitution expression is set.

3 Cancel the job.

***CANCEL:** The procedure is canceled as if a // CANCEL statement had been processed.

***GOTO-label:** Control continues at the TAG label specified. This value must be followed by a label of up to 8 characters. If the specified label is not found in the procedure, message SYS3783 (TAG statement missing) is issued.

DFTACN

Specifies the default action that is taken for escape messages issued by CL commands in procedures running in the System/36 environment. The default action is taken for any message that is not in the message list and when there is no message list. The default action is not used if the message list contains a message ID of *ANY.

***SAME:** The value does not change.

***CONTINUE:** Processing continues with the next statement after the CL command. The ID of the escape message is saved and can be retrieved by the ?MSGID? substitution expression.

***IGNORE:** Processing continues with the next statement after the CL command. The ID of the escape message is not saved, and the ?MSGID? substitution expression is null.

***HALT-options:** A halt with options is issued. This value may be followed by a list of options allowed on the halt. If the options to be allowed are not specified, options 0 (continue) and 3 (cancel) are allowed.

A list of options is a value consisting of up to 4 numbers ranging from 0 to 3, each of which represents an option that is allowed in response to the halt. The meaning of each possible value in the list follows:

0 Continue. The ?MSGID? substitution expression is set.

1 Retry the command. The ?MSGID? substitution expression is not set.

2 Cancel the job step. The ?MSGID? substitution expression is set.

3 Cancel the job.

***CANCEL:** The procedure is canceled as if a // CANCEL statement had been processed.

***GOTO-label:** Control continues at the TAG label specified. This value must be followed by a label of up to 8 characters. If the specified label is not found in the procedure, message SYS3783 (TAG statement missing) is issued.

SCOPE

Specifies the scope of the message list and default action entered on the command.

***CURPRC:** The message list and default action apply only to the procedure in which the command is placed. It is not propagated to lower level procedures, and it is not used after the procedure ends. This value can be entered only in a procedure.

***PRVPRC:** The message list and default action apply only to the procedure that called the procedure in which the command is placed. This value can be entered only in a procedure below first-level.

***JOB:** The default action applies to all procedures in the current System/36 job. The action takes effect immediately, and remains in effect until the first-level procedure ends. The default action specified applies to all procedures in the current job that do not have a default action set.

***SESSION:** The default action applies to all procedures run in the current session. It takes effect immediately, and remains in effect until the user signs off, enters the ENDS36 command, or issues another CHGS36MSGL command with SCOPE(*SESSION) specified.

Examples

Example 1: Setting Up a Message List

```
CHGS36MSGL  MSGL((CPF9801 *GOTO NOTEXIST)
  ((CPF9802 CPF9820) *GOTO NOTAUT)  (*ANY *HALT 3))
CHKOBJ ?2?/?1? *PGM
// GOTO OK
// TAG NOTEXIST
  (code to handle object does not exist messages)
// GOTO OK
// TAG NOTAUT
  (code to handle not authorized to object messages)
// TAG OK
CHGS36MSGL MSGL(*NONE)
```

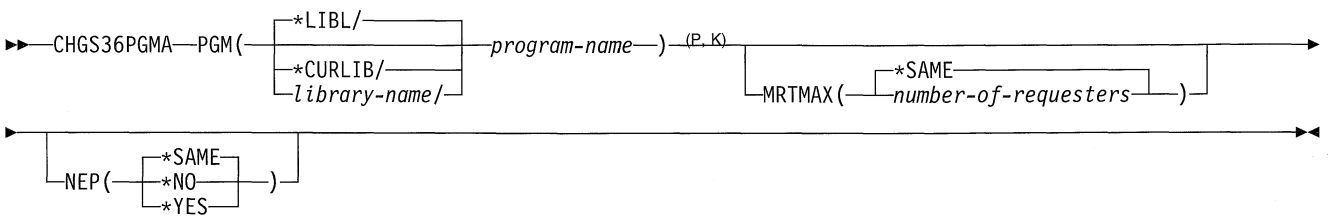
This command sets up a message list to go to label NOTEXIST if message CPF9801 is issued, and to label NOTAUT if either message CPF9802 or CPF9820 is issued. If any other message is entered, a halt with only option 3 (cancel) is issued. The second CHGS36MSGL command removes the message list.

Example 2: Setting the ?MSGID? Substitution Expression

```
CHGS36MSGL MSGL((CPF2105 *IGNORE) (*ANY *CONTINUE))
DLTF ?FLIB?/?1?
// IFF ?MSGID?/ ... (handle error)
```

In this example, message CPF2105 (object not found) is ignored; that is, the ?MSGID? substitution expression is not set. For any other messages, the ?MSGID? substitution expression is set to the message ID. The procedure is attempting to delete a file that may or may not exist. Because the object not found exception is not considered an error in this case, it is ignored. Any other message is handled as an error.

CHGS36PGMA (Change System/36 Program Attributes) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change System/36 Program Attributes (CHGS36PGMA) command changes the attributes of the specified program.

Required Parameter

PGM

Specifies the qualified name of the program having its attributes changed.

The name of the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the program.

Optional Parameters

MRTMAX

Specifies the maximum number of multiple requester terminals that can be attached to the program.

***SAME:** The value does not change.

number-of-requesters: Specify the maximum number of requesters for a program. Valid values range from 1 through 256. The value cannot be increased beyond the current value.

NEP

Specifies whether the program is a never-ending program (NEP). NEP is defined as a long-running program.

***SAME:** The value does not change.

***NO:** The program is not a never-ending program.

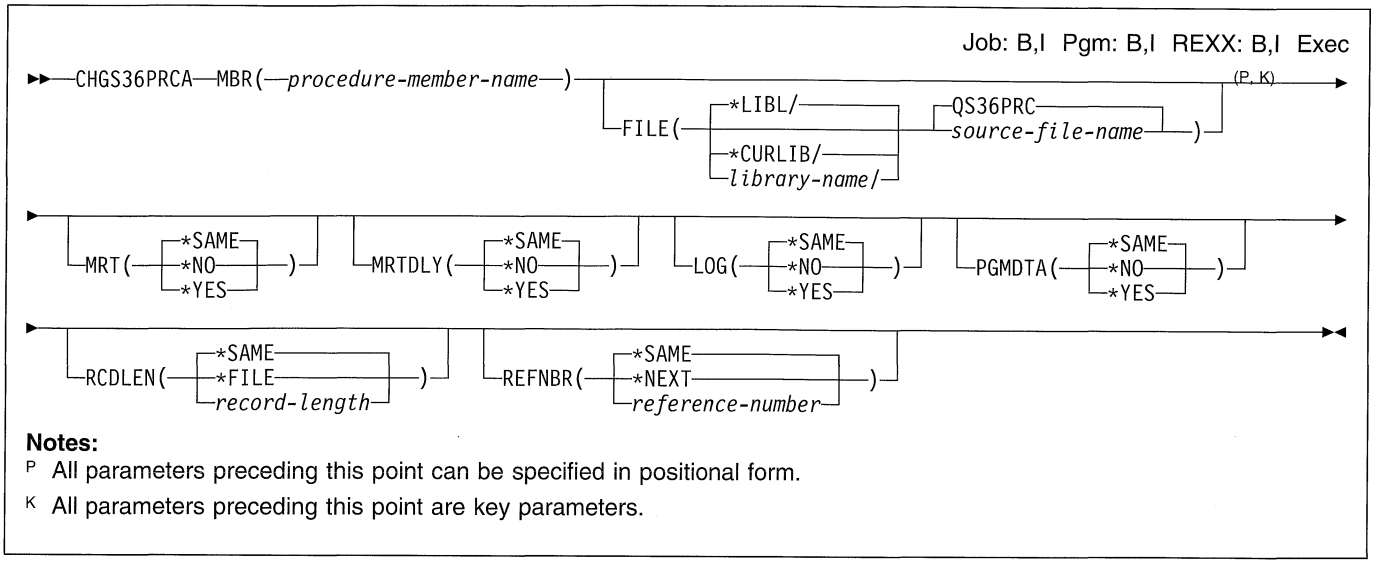
***YES:** The program is a never-ending program.

Example

```
CHGS36PGMA  PGM(RPGLIB/RPGPGM)  MRTMAX(3)  NEP(*YES)
```

This command changes program RPGPGM in RPGLIB to allow up to three MRTs and to be a never-ending program.

CHGS36PRCA (Change System/36 Procedure Attributes) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change System/36 Procedure Attributes (CHGS36PRCA) command changes the attributes of the specified procedure.

Required Parameter

MBR

Specifies the name of the procedure member that will have its attributes changed.

Optional Parameters

FILE

Specifies the qualified name of the physical file containing the procedure member.

The name of the physical file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

QS36PRC: The name of the default source physical file that contains the procedure member.

source-file-name: Specify the name of the source physical file containing the procedure member.

MRT

Specifies whether the procedure is a multiple requester terminal (MRT) procedure.

***SAME:** The value does not change.

***NO:** The procedure is not a multiple requester terminal procedure.

***YES:** The procedure is a multiple requester terminal procedure.

MRTDLY

Specifies whether the procedure uses the default MRT delay value.

***SAME:** The value does not change.

***NO:** The procedure does not use the default MRT delay value.

***YES:** The procedure uses the default MRT delay value.

LOG

Specifies whether the OCL statements are logged to the job log.

***SAME:** The value does not change.

***NO:** The OCL statements are not logged to the job log.

***YES:** The OCL statements are logged to the job log.

PGMDTA

Specifies whether the procedure passes inline data to a program.

***SAME:** The value does not change.

***NO:** The procedure passes parameters to the program.

***YES:** The procedure does not pass inline data to a program.

RCDLEN

Specifies the logical record length of the statements in the procedure member (used by the Save System/36 Library Member (SAVS36LIBM) command).

***SAME:** The value does not change.

***FILE:** The maximum record length of the file containing the procedure member is used.

record-length: Specify the logical record length of the statements in the member. Valid values range from 40 through 120 and must be less than or equal to the record length of the file.

REFNBR

Specifies the reference number assigned to the procedure member.

***SAME:** The value does not change.

***NEXT:** The current reference number is incremented by one. If the current reference number is 999999, then the reference number is returned to 0.

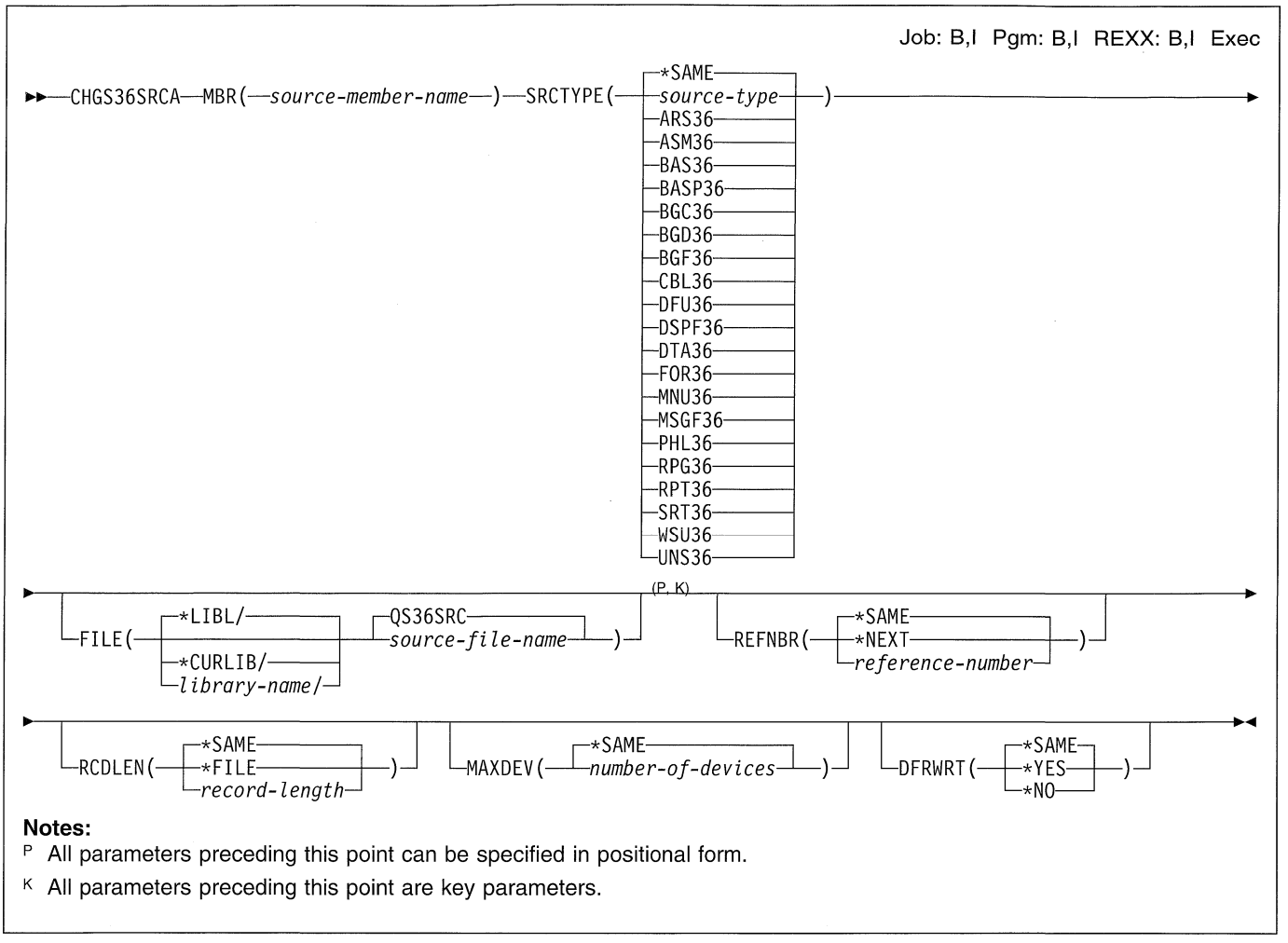
reference-number: Specify the reference number of the procedure member. Valid values range from 0 through 999,999.

Example

```
CHGS36PRCA  MBR(RPGPROC)  FILE(RPGLIB/QS36PRC)
  MRT(*YES)  RCDLEN(*FILE)  REFNBR(*NEXT)
```

This command changes procedure RPGPROC in file QS36PRC in library RPGLIB to be an MRT procedure with a logical record length the same as the QS36PRC file, and increments the current reference number by one.

CHGS36SRCA (Change System/36 Source Attributes) Command



Notes:

P All parameters preceding this point can be specified in positional form.

K All parameters preceding this point are key parameters.

Purpose

The Change System/36 Source Attributes (CHGS36SRCA) command changes the attributes of the specified source member.

Required Parameters

MBR

Specifies the name of the source member whose attributes are being changed.

SRCTYPE

Specifies the source type of the source member.

***SAME:** The value does not change.

source-type: Specify the value for the source type of the member.

Value	Type of member
ARS36	Automatic response member
ASM36	Assembler member
BAS36	BASIC member
BASP36	BASIC procedure (source member)
BGC36	Business graphics utility chart member
BGD36	Business graphics utility data member
BGF36	Business graphics utility format member
CBL36	COBOL member
DFU36	Data file utility member
DSPF36	Display format member
DTA36	Data member
FOR36	FORTRAN member
MNU36	Menu member
MSGF36	Message member
PHL36	Phone list member
RPG36	RPG member
RPT36	RPG auto report member
SRT36	Sort member

WSU36 Work station utility member

UNS36 Unspecified

Optional Parameters

FILE

Specifies the qualified name of the source physical file that contains the source member.

The name of the physical file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

QS36SRC: The name of the default source physical file that contains the source member.

source-file-name: Specify the name of the source physical file that contains the source member.

REFNBR

Specifies the reference number assigned to the source member.

***SAME:** The value does not change.

***NEXT:** The current reference number is increased by one. If the current reference number is 999,999, then the reference number returns to 0.

reference-number: Specify the reference number of the source member. Valid values range from 0 through 999,999.

RCDLEN

Specifies the logical record length of the statements in the source member (used by the Create System/36 Message File (CRTS36MSGF), Create System/36 Menu (CRTS36MNU), and Save System/36 Library Members (SAVS36LIBM) commands).

***SAME:** The value does not change.

***FILE:** The record length of the file specified by the FILE parameter is used.

record-length: Specify the logical record length of the statements in the member. Valid values range from 40 through 120 and must be less than or equal to the record length of the file.

MAXDEV

Specifies the maximum number of devices for a SFGR display file (used by the Create System/36 Display File (CRTS36DSPF) command).

***SAME:** The value does not change.

number-of-devices: Specify the maximum number of devices for a source member. Valid values range from 1 through 256.

DFRWRT

Specifies that the writing of data to the display file is delayed until a read request is made. This parameter is used by the CRTS36DSPF (Create System/36 Display File) command when creating a display file from SFGR source located in this member. Control is returned to the requesting program immediately after the data is received for output. This may result in improved performance.

***SAME:** The value does not change.

***YES:** When a write request is made to the display file, control is returned after the buffer is processed. The data might not be shown immediately; the actual display of the data might take place later when a read or combined write/read operation is performed. The buffer is then available for the next read or combined write/read operation.

***NO:** After a write operation to the display file, the entire input/output request is completed before returning control to the requesting program. All input/output feedback information will be available.

Examples

Example 1: Specifying Maximum Devices and Record Length

```
CHGS36SRCA  MBR(SFGRSRC)  FILE(SDALIB/QS36SRC)
  REFNBR(*NEXT)  RCDLEN(80)  MAXDEV(5)
```

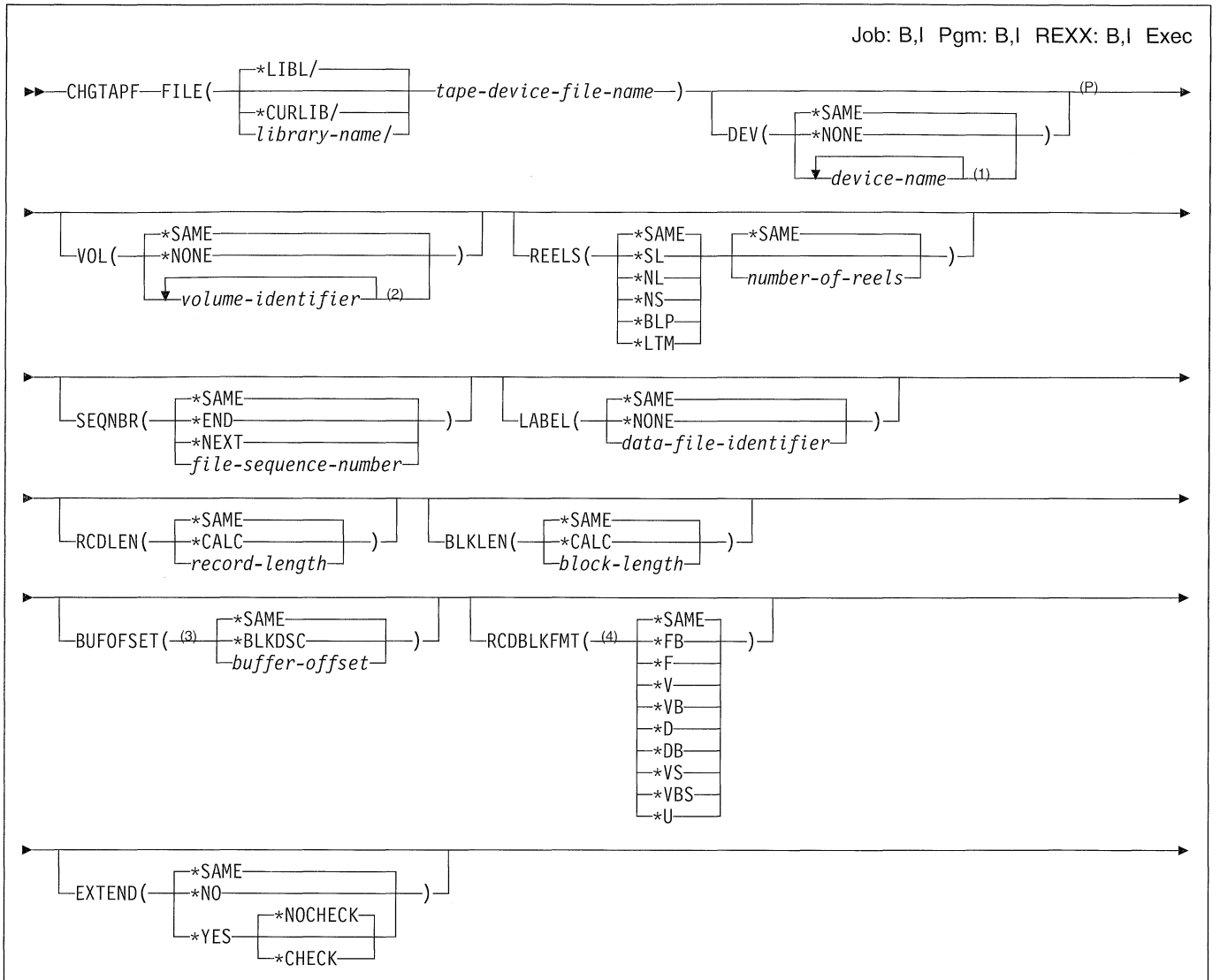
This command changes source member SFGRSRC in file QS36SRC in library SDALIB to allow up to five devices and to have a record length of 80. It also increases the current reference number by one.

Example 2: Turning Off the Defer Write Attribute

```
CHGS36SRCA  MBR(SFGRSRC)  FILE(SDALIB/QS36SRC)
  REFNBR(*NEXT)  RCDLEN(80)  MAXDEV(5)  DFRWRT(*NO)
```

This command changes source member SFGRSRC in the file QS36SRC in library SDALIB to allow up to five devices and to have a record length of 80. It also increases the current reference number by one and turns off the defer write attribute.

CHGTAPF (Change Tape File) Command



Notes:

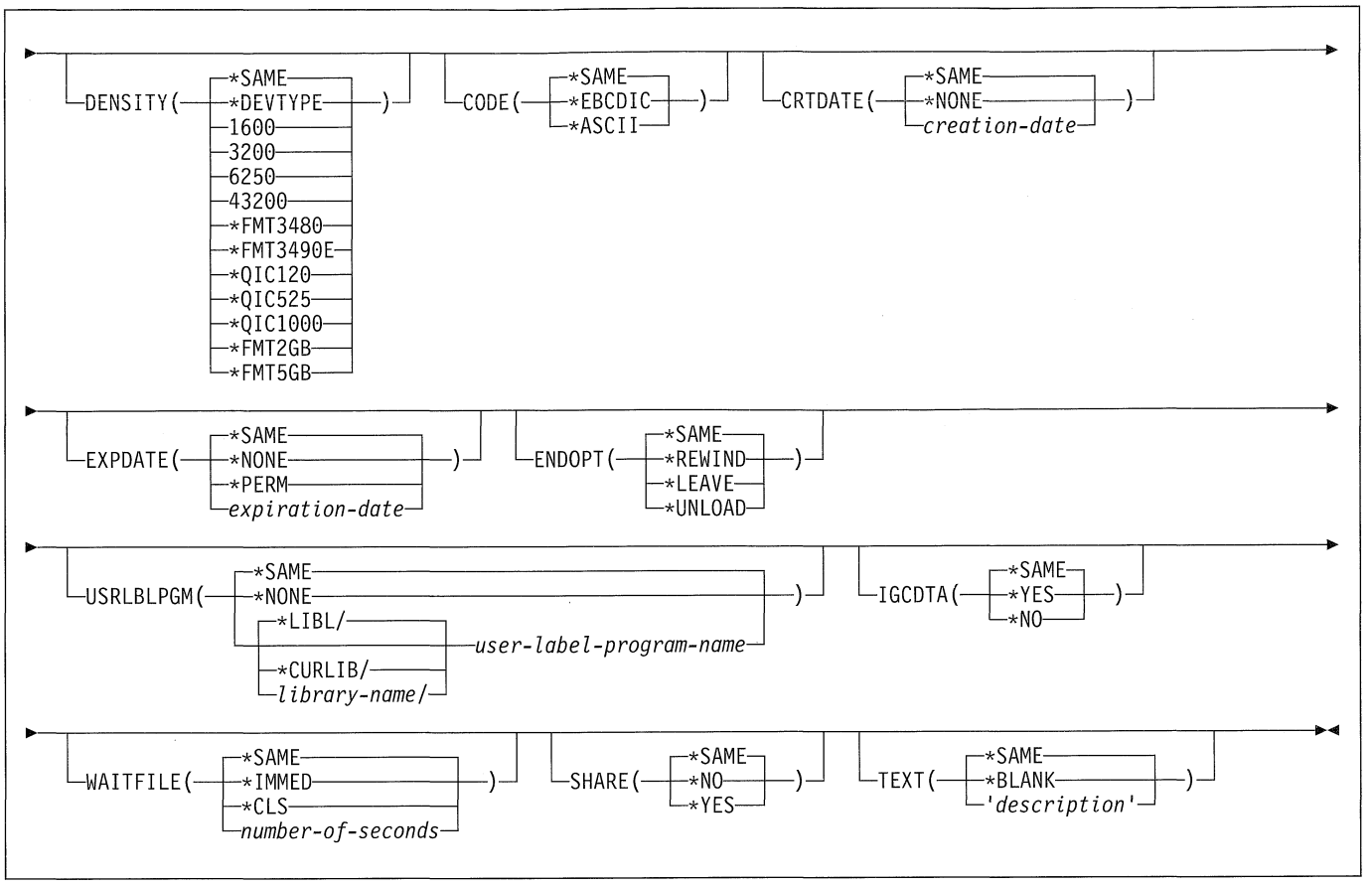
1 A maximum of 4 repetitions

P All parameters preceding this point can be specified in positional form.

2 A maximum of 50 repetitions

3 The value *BLKDSC is valid only if the file is record block format *D or *DB.

4 The values *F, *FB, *VS, *VBS and *U are valid for both EBCDIC and ASCII codes; *V and *VB are valid only for EBCDIC; *D and *DB are valid only for ASCII.



Purpose

The Change Tape File (CHGTAPF) command changes, in the file description, one or more of the attributes of the specified tape device file.

Restriction: Non-labeled tapes cannot be duplicated to 1/4 inch or 8mm cartridge devices.

Required Parameter

FILE

Specifies the qualified name of the tape device file whose description is being changed.

The name of the tape device file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

tape-device-file-name: Specify the name of the file being changed.

Optional Parameters

DEV

Specifies the names of one or more tape devices used with this tape device file to perform input/output data operations.

***SAME:** The value does not change.

***NONE:** No device name is specified. The name of the tape device must be specified later in a CHGDSPF or OVRDSPF command, or in the HLL program that opens the file.

device-name: Specify the names of one or more devices (no more than four) used with this tape device file. The order in which the device names are specified is the order in which tapes on the devices are processed. When more volumes are processed than the number of devices in the DEV list, the devices are used in the same order specified, wrapping around to the first device as needed. Each device name must be known on the system by a device description before this device file is created.

VOL

Specifies one or more volume identifiers used by the file. The volumes must be installed in the same order as the identifiers are specified here (and as they are specified on the DEV parameter). If the file is opened for read backward, then the volume identifiers in the list are processed from last to first (while the devices in the device

list are used in first-to-last order). If a list of volume identifiers is provided for the file, operator messages indicate the name of the required volume. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***NONE:** No tape volume identifiers are specified for this file. They can be supplied before the device file is opened, either in a CHGTAPF or OVRTAPF command or in the high-level language program. If volume identifiers are not specified before the device file is opened, volume checking is not performed beyond verifying that the correct label type volume is on the device, and volume names are not provided in operator messages. The maximum number of reels processed for an *NL, *NS, *BLP, or *LTM input file when VOL(*NONE) is specified is determined by the REELS(number-of-reels) parameter value.

volume-identifier: Specify the identifiers of one or more volumes in the order in which they are placed on the device. Each volume identifier contains a maximum of 6 alphanumeric characters. Use a blank as a separator character when listing multiple identifiers. Up to 50 volume identifiers can be specified. These identifiers are used in messages sent to the operator during processing. The maximum number of reels processed for an *NL, *NS, *BLP, or *LTM input file is determined by the number of volume identifiers in the list.

Note: If the VOL parameter value used for the file specifies a list of identifiers rather than VOL(*NONE), the number-of-reels part of the REELS parameter is ignored regardless of where it is specified. A description of how the parameter values for the file are determined when overrides are used, the high-level language interface, and the device file when the file is opened is in the *Data Management Guide*. To ensure that the number-of-reels part of the REELS parameter is used (rather than a VOL identifier list) to control the volumes processed by the tape device file, specify VOL(*NONE) in the same command in which the REELS parameter is specified.

REELS

Specifies the type of labeling used on the tape reels and the maximum number of reels processed if both a list of volume identifiers is not specified (VOL parameter) and this device file is used with either *NL, *NS, *LTM, or *BLP input files. When the number of reels is specified as the second element of this parameter, the volume identifiers on the volumes are ignored if labeled tapes are being processed; instead, the order in which the reels are installed on the device must be checked by the operator.

The number-of-reels value is not a limiting value for standard-label or output files. For a standard-label *input* file, the data file labels limit the number of volumes processed by indicating end-of-file. For an *output* file, the number-of-reels value is ignored; the system requests that additional volumes be kept on the device until the file is closed.

The system checks the first record following the load point on the tape to see (1) whether it has exactly 80 bytes for EBCDIC or at least 80 bytes for ASCII and (2) whether the first 4 bytes contain the values VOL and 1. If so, the reel contains a standard-label tape. *SL and *BLP files require standard-label tape volumes. *NL, *NS, and *LTM tape files cannot process standard-label volumes.

Note: The values *SL, *NL, and *LTM can be specified if the device file is used for either reading or writing on tapes. The values *NS and *BLP are valid only if the device file is used to read tapes.

Element 1: Type of Labels

***SAME:** The value does not change.

***SL:** The volumes have standard labels. If a list of volume identifiers is specified (with the VOL parameter), the system checks that the correct tape volumes are on the device in the specified sequence.

- If no volume identifier list is given and the file is opened for *output*, any standard-label volumes may be installed on the device.
- If no volume identifier list is given and the file is opened for *input*, the first volume may have any volume identifier, but if the file is continued, the system requires the correct continuation volumes to be processed (verified by checking the data file labels). For an input file, the end-of-file message is sent to the program being used when the labels on the last volume processed indicate that it is the last volume for the data file.

***NL:** The volumes are not labeled. On a nonlabeled volume, tape marks are used to indicate the end of each data file and the end of the volume. For an *input* file, the end-of-file message is sent to the program when the number of volumes specified in the volume list have been processed, or, if no list of volume identifiers is provided, when the number of reels specified in the REELS parameter are processed.

***NS:** The volumes have nonstandard labels. Each volume must start with some kind of label information, optionally preceded by a tape marker and always followed by a tape marker. This nonstandard label information is ignored. The system spaces forward to a point beyond the tape marker that follows the nonstandard labels and positions the tape at the file's data. Each reel must have a tape marker at the end of the file's data. Information beyond this ending tape marker is ignored. Only a single data file can exist on a nonstandard tape. Standard-label volumes *cannot* be processed by using the *NS label processing.

For an *input* file, the end-of-file message is sent to the program using the file when the number of volumes

specified in the volume list have been processed, or, if no list of volume identifiers is provided, when the number of reels specified in the REELS parameter are processed.

***BLP:** Standard-label processing is bypassed. Each reel *must* have standard labels. Although each reel is checked for a standard volume label and each file must have at least one standard header label (HDR1) and one standard trailer label (EOV1 or EOF1), most other label information (such as the data file record length or block length) is ignored. The sequence number of each file on the volume is determined only by the number of tape markers between it and the start of tape (in contrast to *SL processing in which the file sequence number stored in the header and trailer labels of each file are used to locate a data file).

Most of the information in the data file trailer label is ignored, but if an end-of-file (EOF) trailer label is found, the end-of-file message is sent to the program using the tape file. If no end-of-file trailer label is encountered by the time the specified number of volumes or reels have been processed (volume identifier list and REELS parameter), the end-of-file message is immediately sent to the program using the tape file. Bypass label processing can be used when the user does not know the name of the file used or when some file label information is incorrect.

***LTM:** The volumes have no labels but do have a single leading tape marker before the first data file. REELS(*LTM) is processed the same as REELS(*NL) except that when SEQNBR(1) is specified for an output file to create the first data file on the tape, a leading tape marker is written at the start of the tape before the first data block.

Element 2: Number of Reels

***SAME:** The value does not change.

number-of-reels: Specify the maximum number of reels to be processed for an *NL, *LTM, *NS, or *BLP input tape operation when a list of volume identifiers is not specified (VOL parameter). If the next reel is not on the device when the end of the currently-processing tape is reached, a message is sent to the operator requesting that the next tape be installed on the next tape device. The number-of-reels value is ignored for a standard-label (*SL) file or for any output file.

SEQNBR

Specifies the sequence number of the data file on the tape being processed.

- When standard-label tapes are used, the four-position file sequence number is read from the first header label of the data file.
- When bypass label processing is used or when standard-label tapes are not used, the system counts the tape markers from the start of the tape to locate the correct sequence number data file to be processed.
- When multiple-file, multiple-volume tapes are processed using REELS(*SL), the file sequence numbers continue consecutively through the volumes; thus, each new data file has a sequence number one greater than the previous file, regardless of its volume location.

***SAME:** The value does not change.

***END:** The file is written on the end of the tape. This value is used only for files that are written to tape.

An error message is shown on the display when a tape device file is used to read from a tape and the *END special value is specified in the tape device file.

***NEXT:** The next file in the sequence is processed. This value is used for files read from tape. If the tape is currently in a position that is prior to the first file, the first file on the tape is processed.

An error message is shown on the display when a tape file is used to write to a tape and the *NEXT special value is specified in the tape file.

file-sequence-number: Specify the sequence number of the file that is used.

LABEL

Specifies the data file identifier of the data file processed by this tape device file. An identifier is defined only for standard-label tapes and is stored in the header label immediately before the data file.

If a data file identifier is specified for any type of label processing other than *SL, it is ignored.

An identifier is required for a standard label output file, but is optional for an input file because the sequence number uniquely identifies the data file to process.

For an input file or output file with EXTEND(*YES) specified, this parameter specifies the identifier of the data file on the tape. The specified identifier must match the one in the labels of the data file that the SEQNBR parameter specifies; otherwise, an error message is sent to the program using this device file. For output files with EXTEND(*NO) specified, this parameter specifies the identifier of the data file to be created on the tape. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***NONE:** The data file identifier is not specified.

data-file-identifier: Specify the identifier (17 alphanumeric characters maximum) of the data file used with this tape device file. If this identifier is for a tape written in the basic exchange format, and is used on a system other than an AS/400 system, up to eight characters or a qualified identifier having no more than eight characters per qualifier must be used.

RCDLEN

Specifies, in bytes, the length of the records contained in the data file processed with this device file. The system

always uses the record length and block length specified in the data file labels for any standard-label input file or output file with EXTEND(*YES) specified (if a second header label (HDR2) is found on the tape and *BLP label processing has not been specified).

***SAME:** The value does not change.

***CALC:** No record length is specified for the data file being processed. If *CALC is specified, the system will attempt to calculate an appropriate record length when the file is opened. RCDLEN(*CALC) can be used for nonlabeled tapes or when there is no HDR2 label if a BLKLEN value other than *CALC is specified for the file and RCDBLKFMT does not specify spanned or blocked records. In this case, the system calculates an appropriate record length from the block length, record block format, and buffer offset (for an ASCII file) specified for the file. In any other case, the actual record length must be specified by a CHGTAPF command or OVRTAPF command, or in the high-level language program that opens the device file.

record-length: Specify a value ranging from 1 through 32767 bytes that indicates the length of each record in the data file. The minimum and maximum record length allowed for a file is dependent on the record block format, block length, buffer offset (for an ASCII file), and recording code. The *Record Length Values* table (at the end of this command description) shows the minimum and maximum record length values allowed for each record block format, assuming the block length value is large enough to support the maximum record length.

BLKLEN

Specifies, in bytes, the maximum length of the data blocks transferred to or from the tape for output or input operations. The system uses the block length and record length specified in the data file labels for any standard-label input file or output file with EXTEND(*YES) specified (if a second header label (HDR2) is found on the tape and *BLP label processing has not been specified).

***SAME:** The value does not change.

***CALC:** No block length is specified for the data file to be processed. If *CALC is specified, the system attempts to calculate an appropriate block length when the file is opened. BLKLEN(*CALC) can be used for nonlabeled tapes or when there is no HDR2 label if a RCDLEN value other than *CALC is specified for the file and RCDBLKFMT does not specify spanned or blocked records. In this case, the system calculates an appropriate block length from the record length, record block format, and buffer offset (for an ASCII file) specified for the file. In any other case, the actual block length must be specified by a CHGTAPF command or OVRTAPF command, or in the high-level language program that opens the device file.

block-length: Specify a value, not exceeding 32767 bytes, that specifies the maximum length of each block in the data file to be processed. The minimum block length that can be successfully processed is determined by the tape device hardware and AS/400 system machine support functions.

- The maximum block length is always 32767 bytes for an input file, but is limited to 9999 bytes if block descriptors must be created for an ASCII output file.
- The following table shows the minimum and maximum block length values allowed for an output file:

Table 17. Absolute Block Length (BLKLEN) Ranges

CODE	BUFOFSET	Minimum BLKLEN	Maximum BLKLEN
*EBCDIC	Ignored	18	32767
*ASCII	0	18	32767
*ASCII	*BLKDSC	18	9999

BUFOFSET

Specifies the buffer offset value for the start of the first record in each block in the tape data file. A buffer offset value can be used for any record block format ASCII file, and is ignored for an EBCDIC tape file. The system uses the buffer offset specified in the data file labels for any standard-label input file or output file with EXTEND(*YES) specified if a value is contained in the second header label (HDR2) on the tape, and *BLP label processing has not been specified.

The buffer offset parameter specifies the length of any information that precedes the first record in the block. For record block formats *D, *DB, *VS, and *VBS, each record or record segment is preceded by a descriptor that contains the length of the record or segment. A buffer offset value is used to indicate that there is information *ahead* of the descriptor word for the first record in each block, or *ahead* of the data of the first fixed-length record or undefined format record in each block.

This parameter is not needed for a standard-label file processed for input if the tape includes a second file header label (HDR2) that contains the buffer offset value. A buffer offset value must be provided by the Create Tape File (CRTTAPF) command, Change Tape File (CHGTAPF) command, or Override Tape File (OVRTAPF) command, or by the file labels for an input file that contains any information (such as a block descriptor) ahead of the first record in each block. If the user does not specify a buffer offset value when a tape file is created, it is not necessary to specify an offset value when the file is read.

The only buffer offset values allowed for an output file are zero and *BLKDSC. An existing standard-label data file with a buffer offset value in the HDR2 label can be extended only if the buffer offset value is either 0 or 4. A buffer offset value of 0 in the HDR2 label adds data blocks with *no* buffer offset. BUFOFSET(*BLKDSC) must be specified to extend an existing tape data file that contains an offset value of 4 in the HDR2 label.

***SAME:** The value does not change.

***BLKDSC:** Creates 4-byte block descriptors in any tape file created by using this device file. Any input file read by using this device file should assume 4 bytes of buffer offset information preceding the first record in each data block. This value is valid only for a record block format *D or *DB file. The contents of the buffer offset information of each output data block when BUFOFSET(*BLKDSC) is specified is the actual length of the data block, expressed in zoned decimal format.

buffer-offset: Specify a value ranging from 0 through 99 bytes that specifies the length of the buffer offset information that precedes the first record in each data block.

RCDBLKFMT

Specifies the type and blocking attribute of records in the tape data file being processed.

Record block format *V and *VB records can be processed only for an EBCDIC file; *D and *DB records can be processed only for an ASCII file. If a standard-label tape (label type *SL or *BLP) is being processed and an inconsistent record block format is specified for the volume code, the correct record type is assumed (V or D) for the volume code and a warning message is sent to the program that opens the file. If the record type and code are inconsistent for a nonlabeled volume (label type *NL, *LTM, or *NS), an error message is sent and the file is *not* opened, because there are no labels to verify the correct volume code.

If a valid record length, block length, and buffer offset value (for an ASCII file) are specified for fixed-length records but the block attribute is incorrect, the correct block attribute is assumed (changing record block format *F to *FB or record block format *FB to *F), and a warning message is sent to the program that opens the file.

If a block length is specified that is longer than required to process a maximum length record, then record block format *V, *D, or *VS is changed to *VB, *DB, or *VBS and a warning message is sent to the program that opens the file.

The *Required RCDLEN/BLKLEN/BUFOFSET Relation* table, at the end of this command description, shows the required relationship between the record length, block length, and buffer offset (for ASCII) file parameters for an output file or an input file where the file parameters are not determined from a second file header label (HDR2).

Note: When BUFOFSET(*BLKDSC) is specified for the file, a value of 4 should be used for the BUFOFSET part of any BLKLEN calculations, unless existing file labels on the tape specify a different value.

***SAME:** The value does not change.

***FB:** Fixed-length, blocked, unspanned records in either EBCDIC or ASCII code are processed. The system may change this record block format to *F, based on other file parameters.

***F:** Fixed-length, unblocked, unspanned records in either EBCDIC or ASCII code are processed. The system may change this record block format to *FB, based on other file parameters.

***V:** Variable-length, unblocked, unspanned records in EBCDIC type V format are processed. The system may change this record block format to *VB, *D, or *DB, based on other file parameters.

***VB:** Variable-length, blocked, unspanned records in EBCDIC type V format are processed. The system may change this record block format to *DB, based on the volume code.

***D:** Variable-length, unblocked, unspanned records in ASCII type D format are processed. The system may change this record block format to *DB, *V, or *VB, based on other file parameters.

***DB:** Variable-length, blocked, unspanned records in ASCII type D format are processed. The system may change this record block format to *VB, based on the volume code.

***VS:** Variable-length, unblocked, spanned records in either EBCDIC or ASCII code are processed. The system may change this record block format to *VBS, based on other file parameters. Note that the representation of spanned records on the tape is different for EBCDIC and ASCII files, but the system selects the correct format based on the file code.

***VBS:** Variable-length, blocked, spanned records in either EBCDIC or ASCII code are processed. Note that the representation of spanned records on the tape differs for EBCDIC and ASCII files, but the system selects the correct format based on the file code.

***U:** Undefined format records in either EBCDIC or ASCII code are processed. RCDBLKFMT(*U) records are processed as variable-length records, and each record written or read is in a separate tape block. This format can be useful for processing tape files that do not have the formatting requirements of any other record block format.

EXTEND

Specifies, for output operations to tape, whether new records are added to the end of a data file currently on the tape. The specific data file is identified by the SEQNBR parameter and, for a standard-label file, the LABEL parameter. If the data file is extended, it becomes the last file on the tape volume; data files that follow it are overwritten as the specified file is extended.

Note: This parameter is not valid for 1/4-inch cartridge tape devices.

***SAME:** The value does not change.

***NO:** Records are not added to the end of the specified data file. If there is already a data file with the specified

SEQNBR on the tape, a new data file is created by overwriting the existing data file and any files that follow it.

Element 1: Adding Records to Data File

***YES:** New records are added to the end of the specified data file on tape when this device file is used.

Element 2: Checking Active Files

***NOCHECK:** The file is extended without being checked to see whether the file is active.

***CHECK:** Before the file is extended, the file is checked to see whether it is active.

DENSITY

Specifies, in bits per inch, the density of the data that is written on the tape volume when this device file is created. This parameter is used only for tape files being written to tape; it is ignored for tape files being read from the tape (in the case of files being read from tape, the density on the tape is used).

The density of a standard-label volume is specified on the INZTAP command, which initializes tapes as standard-label volumes by writing volume labels on them. If the density specified on this parameter is different than the density of a standard-labeled tape, the density specified on this parameter is used, and a warning message is sent.

***SAME:** The value does not change.

***DEVTYPE:** The density is based on the kind of tape device being used.

1600: The data density on this tape volume is 1,600 bits per inch.

3200: The data density on this tape volume is 3,200 bits per inch.

6250: The data density on this tape volume is 6,250 bits per inch.

43200: The data density on this tape volume is 43,200 bits per inch.

***FMT3480:** The data density on this tape volume is formatted to support a 3480 device.

***FMT3490E:** The data density on this tape volume is formatted to support a 3490 device.

***QIC120:** The format of this tape is *QIC120.

***QIC525:** The format of this tape is *QIC525.

***QIC1000:** The format of this tape is *QIC1000.

| ***FMT2GB:** The format of this tape is *FMT2GB.

| ***FMT5GB:** The format of this tape is *FMT5GB.

CODE

Specifies the character code used. The code can be either extended binary-coded decimal interchange code (*EBCDIC) or the American National Standard Code for Information Interchange (*ASCII).

***SAME:** The value does not change.

***EBCDIC:** The extended binary-coded decimal interchange code (EBCDIC) character set code is used.

***ASCII:** The ASCII character set code is used.

CRTDATE

Specifies, for tape input data files and for tape output for which EXTEND(*YES) is specified, the date when the data file was created (written on tape).

Note: The data file creation date is stored in file labels on the tape. If a creation date is specified for any type of label processing other than standard-label (*SL), it is ignored. If the creation date written on the tape containing the data file does not match the date specified in this device file description, an inquiry message is sent to the operator.

***SAME:** The value does not change.

***NONE:** The creation date is not specified. It is not checked unless it is supplied before the device file is opened, either in a OVRTAPF command or CHGTAPF command, or in the high-level language program.

creation-date: Specify the creation date of the data file used by this device file. The date must be specified in the format defined by the job attributes DATFMT and, if separators are used, DATSEP.

EXPDATE

| Specifies, for tape output data files only, and only when standard-labeled tapes are used, the expiration date of
| the data file used by this device file. If a date is specified, the data file is protected and cannot be overwritten
| until after the specified expiration date. The files cannot
| be overwritten until after the expiration date.

Note: The data file expiration date is stored in file labels on the tape. If an expiration date is specified for any type of label processing other than *SL, it is ignored.

***SAME:** The value does not change.

***NONE:** No expiration date for the data file is specified; the file is not protected. An expiration date is written in the data file labels so the file can be used as a scratch data file.

***PERM:** The data file is permanently protected. An expiration date of 999999 is assigned.

| *expiration-date:* Specify the date on which the data file
| expires, after which it can be overwritten with new data.
| The expiration date must be later than or equal to the
| current date. The date must be specified in the format defined by the job attributes QDATFMT and, if separators are used, QDATSEP.

ENDOPT

Specifies the operation that is automatically performed on the tape volume after the operation ends. If more than one volume is included, this parameter applies only to the last tape volume used; all other tape volumes are

rewound and unloaded when the end of the tape is reached.

***SAME:** The value does not change.

***REWIND:** The tape is automatically rewound, but not unloaded, after the operation has ended.

***LEAVE:** The tape does not rewind or unload after the operation ends. It remains at the current position on the tape drive.

This option is used to reduce the time required to position the tape if the next tape file that opens to this device uses a data file that is on this volume.

Note: Even if ENDOPT(*LEAVE) is specified, the next tape file opened to this reel is positioned at the beginning of some data file on the volume (or at the end of a data file, for either read backward or for output that extends an existing data file on the volume). A tape file is always positioned at the start or end of a data file when it is opened.

***UNLOAD:** The tape is automatically rewound and unloaded after the operation ends.

USRLBLPGM

Specifies the qualified name of the user program that processes user tape labels. On an output file, the user label program will pass the user labels that are written to tape. On an input file, the user labels are passed to the user label program.

***SAME:** The value does not change.

***NONE:** There is no user label program for this device file.

The name of the user label program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

user-label-program-name: Specify the name of the user program that processes the user tape labels. If no library qualifier is given, *LIBL is used to find the file.

IGCDTA

Specifies whether the file processes double-byte character set (DBCS) data.

***SAME:** The value does not change.

***YES:** The file processes DBCS data.

***NO:** The file does not process DBCS data.

WAITFILE

Specifies the number of seconds that the program waits for the file resources and session resources to be allocated when the file is opened, or for the device or session resources to be allocated when an acquire operation is performed to the file. If those resources are not allocated within the specified wait time, an error message is sent to the program. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

Note: An immediate allocation of the device by the device resource is required when an acquire operation is performed to the file.

***SAME:** The value does not change.

***IMMED:** The program does not wait; when the file is opened, an immediate allocation of the file resources is required.

***CLS:** The job default wait time is used as the wait time for the file resources being allocated.

number-of-seconds: Specify the number of seconds that the program waits for the file resources to be allocated to the tape file when the file is opened, or the wait time for the device allocated when an acquire operation is performed to the file. Valid values range from 1 through 32767 seconds.

SHARE

Specifies whether the open data path (ODP) for the tape file is shared with other programs in the routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

More information on shared database files is in the *Database Guide*.

***SAME:** The value does not change.

***NO:** The ODP created by the program with this attribute is not shared with other programs in the routing step. Every time a program opens the file with this attribute, a new ODP to the file is created and activated.

***YES:** The ODP created with this attribute is shared with each program in the routing step that also specifies SHARE(*YES) when it opens the file.

Note: When SHARE(*YES) is specified and control is passed to a program, a read operation in that program retrieves the next input record. A write operation produces the next output record.

TEXT

Specifies text that briefly describes the tape device file. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Table 18. Record Length Values

Absolute RCDLEN Ranges					
		FILETYPE(*DATA)		FILETYPE(*SRC)	
CODE	RCDFBLKFMT	Minimum RCDLEN	Maximum RCDLEN	Minimum RCDLEN	Maximum RCDLEN
*EBCDIC *ASCII	*F *FB *U *F *FB *U	18 18	32767 32767	30 30	32767 32767
*EBCDIC *ASCII	*V *VB *D *DB	1 1	32759 9995	13 13	32767 10007
*EBCDIC *ASCII	*VS *VBS *VS *VBS	1 1	32759 32759	13 13	32767 32767

Table 19. Required RCDLEN/BLKLEN/BUFOFSET Relation

CODE	RCDBLKFMT	BLKLEN = fcn(RCDLEN,BUFOFSET)
*EBCDIC *ASCII	*F *U *F *U	BLKLEN = RCDLEN BLKLEN=RCDLEN + BUFOFSET
*EBCDIC *ASCII	*FB *FB	BLKLEN = RCDLEN * n BLKLEN = (RCDLEN * n) + BUFOFSET n is the number of records in a maximum-length block
*EBCDIC *ASCII	*V *D	BLKLEN = RCDLEN * 8 BLKLEN = RCDLEN * 4 + BUFOFSET
*EBCDIC *ASCII	*VB *DB	BLKLEN >= RCDLEN + 8 BLKLEN >= RCDLEN + 4 + BUFOFSET
*EBCDIC *ASCII	*VS *VBS *BS *VBS	BLKLEN >= 18 BLKLEN >= 6 + BUFOFSET (18 minimum)

Examples

Example 1: Changing the Tape File Description

```
CHGTAPF   FILE(TAPE01)  LABEL(TUESDAY)
```

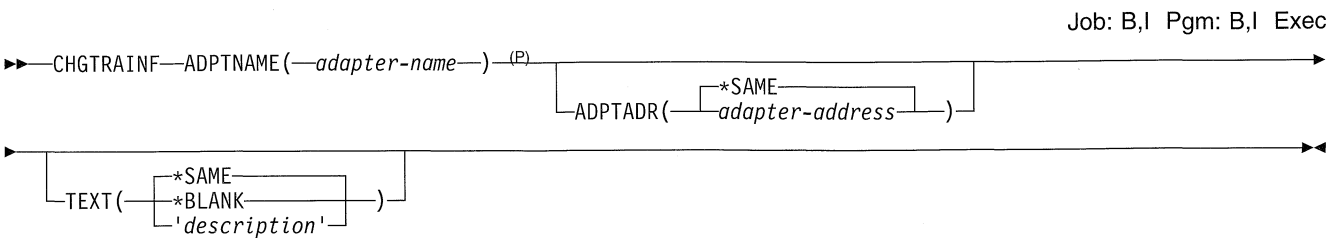
This command changes the description of the tape device file named TAPE01. The LABEL parameter now contains the data file identifier TUESDAY.

Example 2: Enabling a Tape File to Process DBCS Data

```
CHGTAPF   FILE(IGCLIB/IGCTAP)  IGCDTA(*YES)
```

This command changes the tape file IGCTAP, which is stored in the library IGCLIB, so that the file processes double-byte character set data.

CHGTRAINF (Change TRLAN Adapter Information) Command



Note:

| P All parameters preceding this point can be specified in positional form.

Purpose

The Change Token-Ring Local Area Network Adapter Information (CHGTRAINF) command changes an adapter name entry in the adapter file.

Required Parameter

ADPTNAME

Specifies the name of the adapter being changed in the adapter file.

Optional Parameters

ADPTADR

Specifies the 12-character hexadecimal adapter address.

***SAME:** The value does not change.

adapter-address: Specify the adapter address being changed.

TEXT

Specifies text that briefly describes the TRLAN adapter.
| More information on this parameter is in Appendix A,
| "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

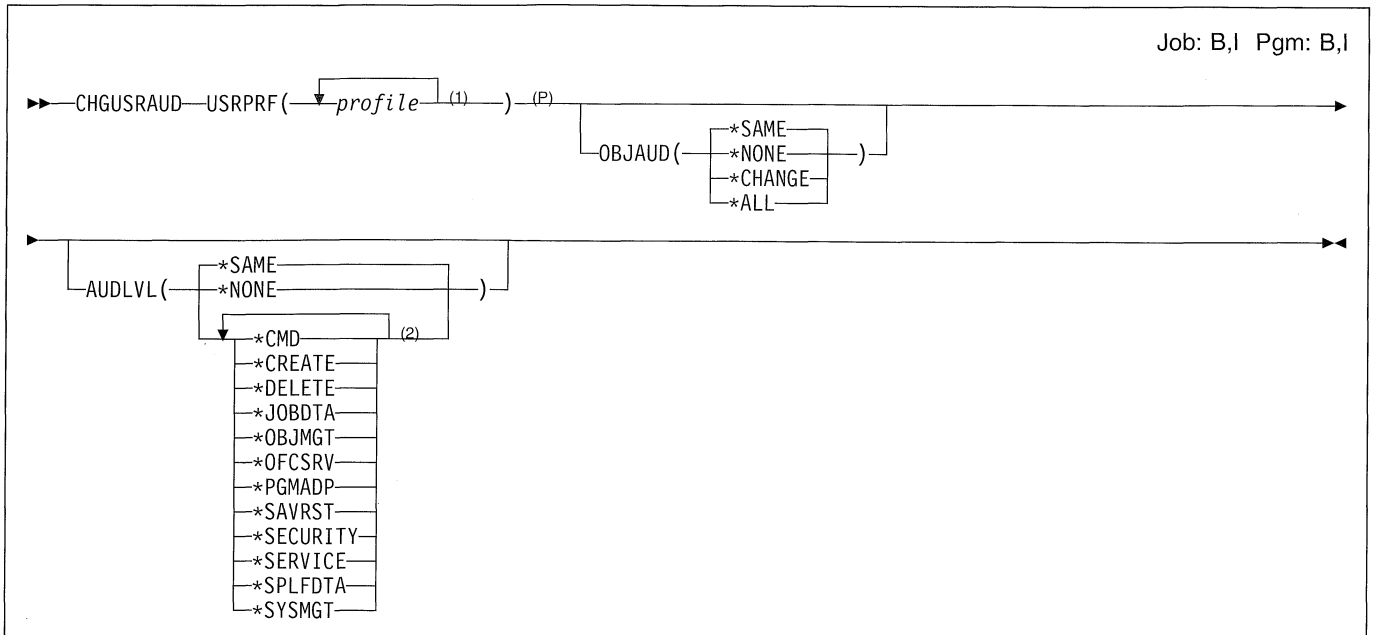
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGTRAINF  ADPTNAME(ENGINEERING)  ADPTADR(00000000110F)
```

This command changes the address of adapter ENGINEERING to 0000000110F.

CHGUSRAUD (Change User Audit) Command



Notes:

1 A maximum of 50 repetitions.

2 A maximum of 12 repetitions.

P All parameters preceding this point can be specified in positional form.

Purpose

The CHGUSRAUD (Change User Audit) command allows a user with *AUDIT special authority to set up or change auditing for a user. The system value QAUDCTL controls turning auditing on and off. The auditing attributes of a user profile can be displayed with the Display User Profile (DSPUSRPRF) command.

Note: The changes made by CHGUSRAUD take effect the next time a job is started for this user.

Required Parameter

USRPRF

Specifies the name of the user profile whose auditing values are changed.

Optional Parameters

OBJAUD

Specifies the object auditing value for the user. This value only takes effect if the object auditing (OBJAUD) value for the object being accessed has the value *USRPRF.

***SAME:** The value does not change.

***NONE:** The auditing value for the object determines when auditing is performed.

***CHANGE:** All change accesses by this user on all objects with the *USRPRF audit value are logged.

***ALL:** All change and read accesses by this user on all objects with the *USRPRF audit value are logged.

AUDLVL

Specifies the level of activity that is audited for this user profile.

Note: The system value QAUDLVL is used in conjunction with this parameter. Example: If QAUDLVL is set to *DELETE and AUDLVL is set to *CREATE, then both *DELETE and *CREATE would be audited for this user. The default value for the QAUDLVL system value is *NONE.

***SAME:** The value does not change.

***NONE:** No auditing level is specified. The auditing level for this user is taken from system value QAUDLVL.

***CMD:** CL command strings, System/36 environment operator control commands, and System/36 environment procedures are logged for this user.

***CREATE:** Auditing entries are sent when objects are created by this user.

***DELETE:** Auditing entries are sent when objects are deleted by this user.

***JOBDTA:** All job start and stop data is audited for this user.

***OBJMGT:** Object management changes made by this user, such as move or rename, are audited.

***OFCSRV:** Office services changes made by this user, such as changes to the system directory and use of OfficeVision/400 mail, are audited.

***PGMADP:** Authority obtained through program adoption is audited for this user.

***SAVRST:** Save and restore actions performed by this user are audited.

***SECURITY:** Security changes made by this user are audited.

***SERVICE:** Use of the system service tools by this user is audited.

***SPLFDTA:** Spool files operations made by this user are audited.

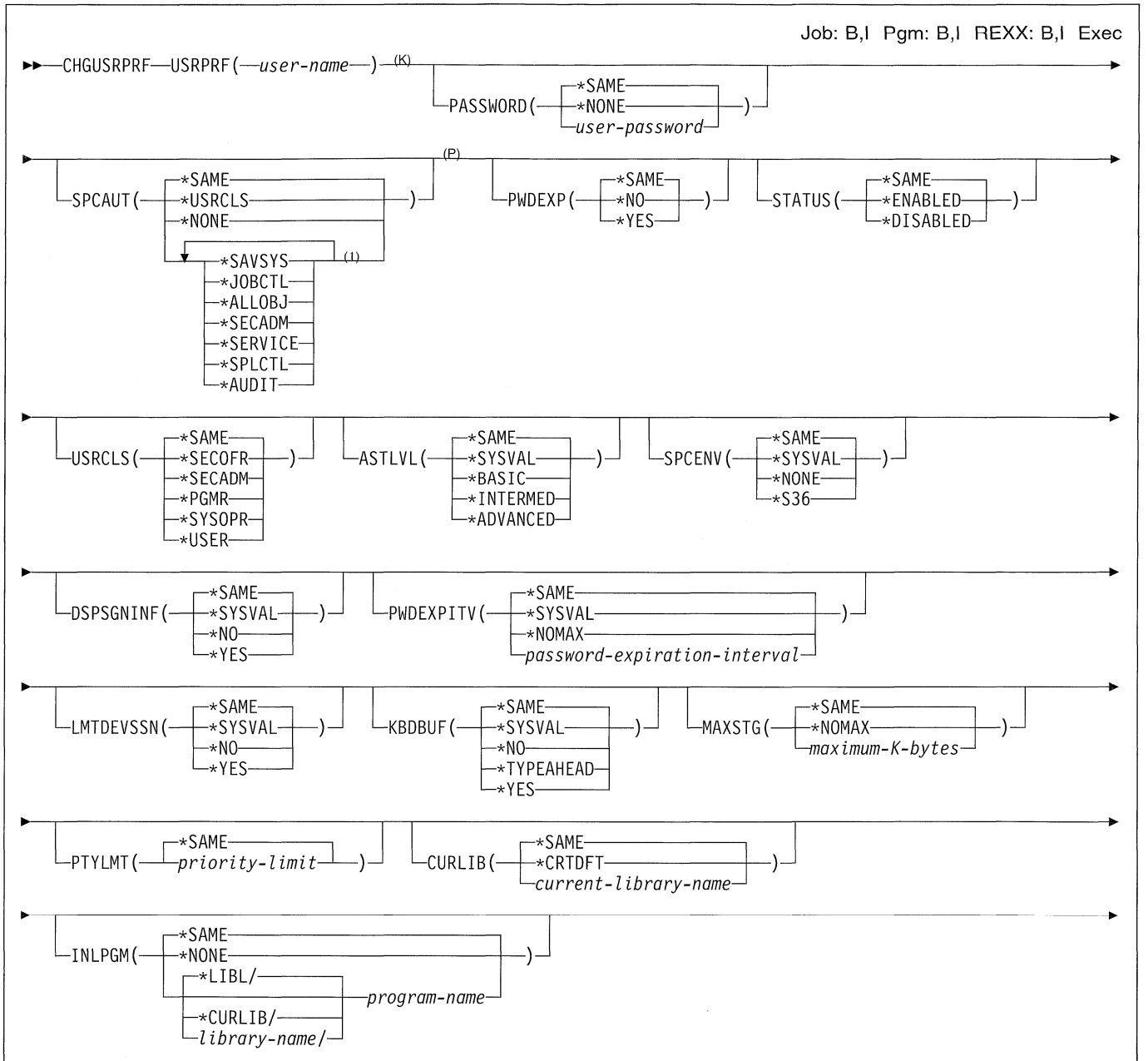
***SYSMGT:** Use of system management functions by this user is audited.

Example

```
CHGUSRAUD  USRPRF(FRED)  OBJAUD(*CHANGE)
  AUDLVL(*CREATE *DELETE)
```

This command changes the auditing value in the user profile of the user FRED. All objects whose object auditing value is *USRPRF are audited when they are changed by user FRED. All objects that are created and all objects that are deleted will be audited for user FRED. Auditing records are sent to the auditing journal QAUDJRN in QSYS.

CHGUSRPRF (Change User Profile) Command

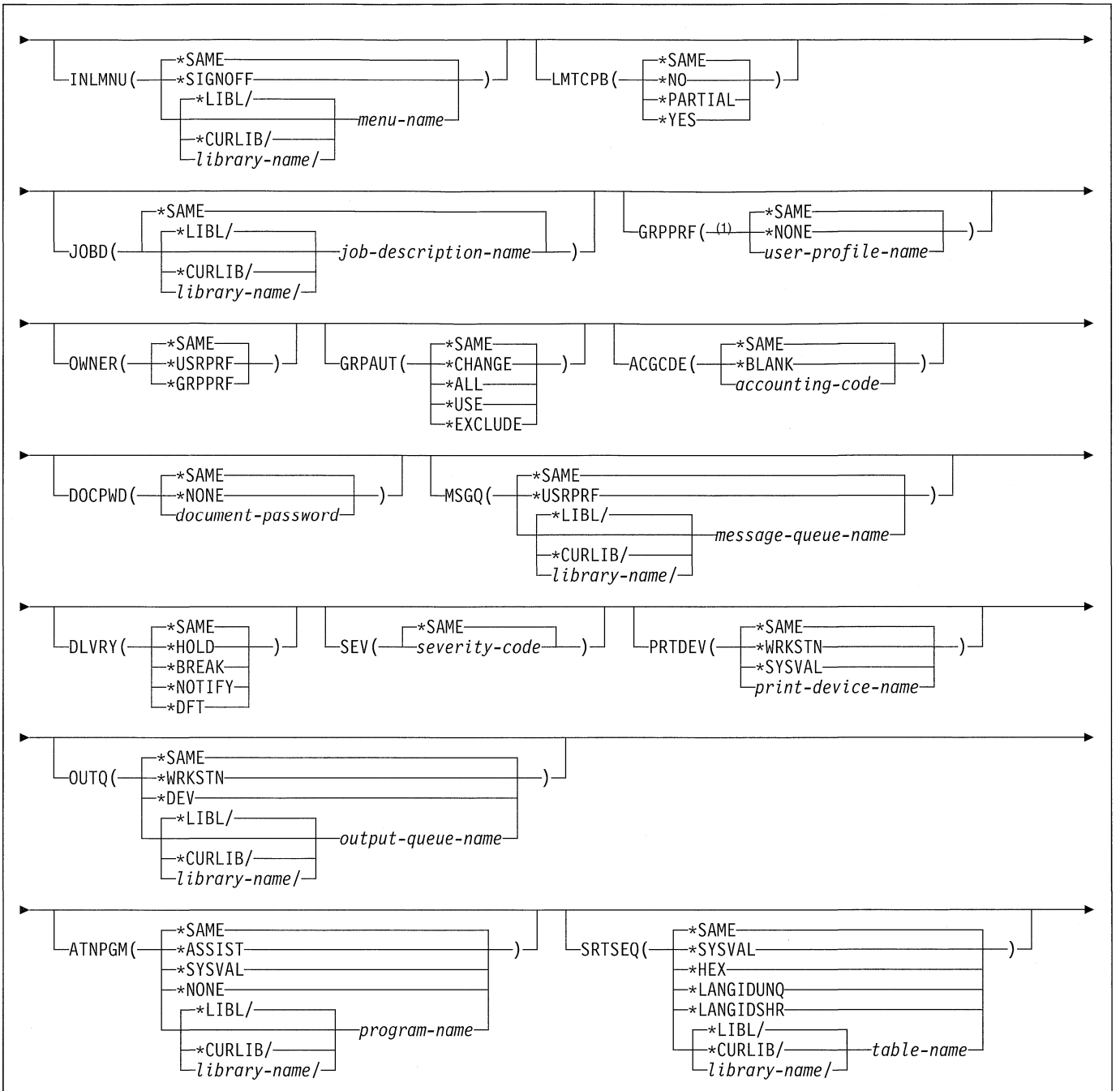


Notes:

K All parameters preceding this point are key parameters.

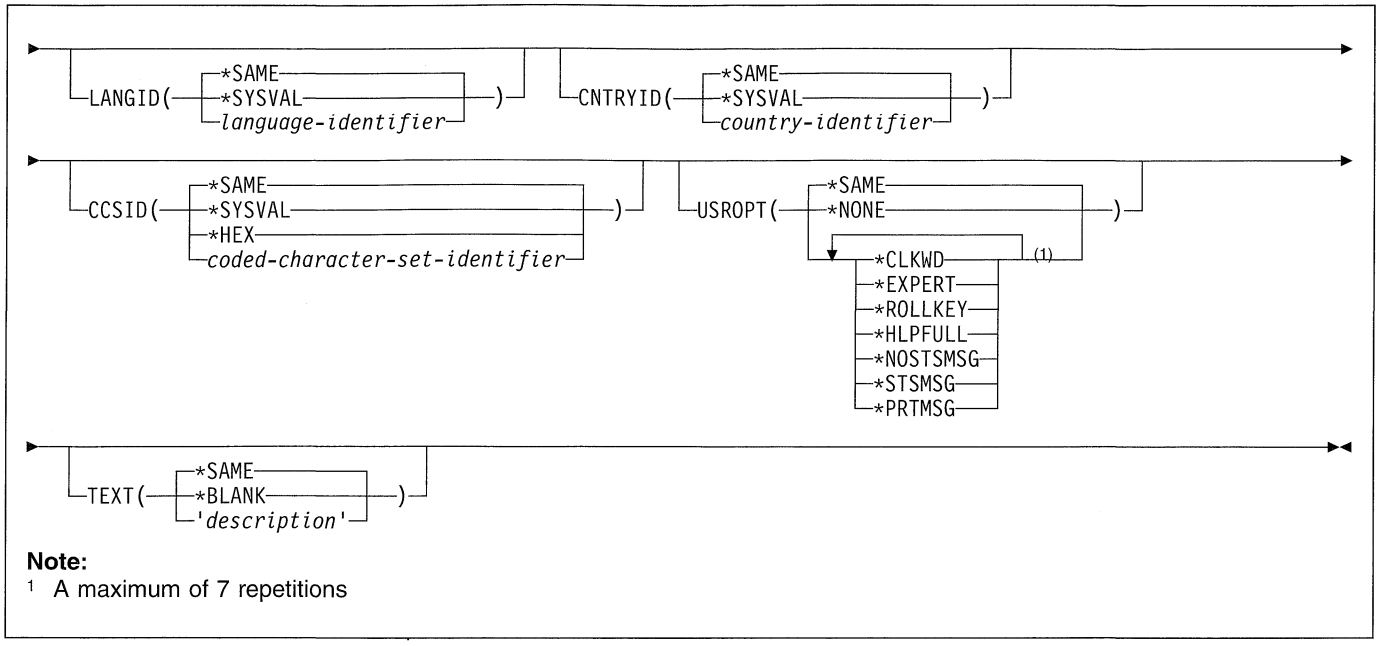
1 A maximum of 6 repetitions

P All parameters preceding this point can be specified in positional form.



Note:

[1] The following user profiles cannot be specified on the GRPPRF parameter: QSLV, QSRVBAS, QRJE, QSYS, QSPL, QDBSHR, QGATE, QSPLJOB, QDOC, QSNADS, QFNC, QLPAUTO, QLPINSTALL, QTSTRQS, QDSNX, and QDFTOWN. The following user profiles cannot have a group profile: QSRV, QSRVBAS, QSECOFR, and QRJE. Because the IBM-supplied user profiles QSYS, QDBSHR, QSPL, QSPLJOB, QDOC, QSNADS, and QFNC are not allowed on the CHGUSRPRF command, they cannot have a group profile.



Note:
[1] A maximum of 7 repetitions

Purpose

The Change User Profile (CHGUSRPRF) command changes the values specified in a user profile.

The password validation rules are not verified by the system when a password is changed by this command. A description of the password validation rules is in *Security Reference*.

Restrictions:

1. Only a user with *SECADM special authority, and *OBJMGT and *USE authorities to the user profile being changed, can specify this command.
2. *USE authority to the current library, program, menu, job description, message queue, print device, output queue, or ATTN key handling program is required to specify these parameters.
3. User profiles QSYS, QDBSHR, QSPL, QDOC, QGATE, QSPLJOB, QLPINSTALL, QTSTRQS, QDFTOWN, QDSNX, QSNADS, QFNC, or QLPAUTO cannot be changed.

Required Parameter

USRPRF

Specifies the name of the user profile whose values are being changed. A numeric user profile can be specified. If the user profile is numeric, then it must begin with a *Q*.

The following IBM-supplied objects are not valid on this parameter:

QDBSHR
QDOC
QDSNX
QFNC
QGATE
QLPAUTO
QLPINSTALL
QDFTOWN
QSNADS
QSPL
QSPLJOB
QSYS
QTSTRQS

Optional Parameters

PASSWORD

Specifies the password that allows the user to sign on the system. The password is associated with a user profile and is used by the system to represent the user in the system. The passwords should be known only to the individual user and to the security officer. A numeric password can be specified. If the password is numeric, then it must begin with a *Q*, for example, *Q1234* where *1234* is the password used for signing on the system.

Note: The new password is not checked against the password validation rules. The password validation rules are defined by AS/400 system values. For a description of the password validation rules, see the *Security Reference*.

***SAME:** The value does not change.

***NONE:** No password is associated with this user profile. Users cannot sign on a system with a profile that has PASSWORD(*NONE) specified.

user-password: Specify the password (an alphanumeric character string of 10 characters or less) that identifies the user with his own user profile. The first character must be alphabetic and the other characters must be alphanumeric. If an all numeric password is used, Q must be the first character.

SPCAUT

Specifies the special authorities given to a user. Special authorities are required to perform certain functions on the system. Special authorities cannot be removed from the QSECOFR or the QSYS user profiles.

The following authorities are usually given:

- Save system authority (*SAVSYS) to users who need to operate the system.
- Job control authority (*JOBCTL) to users who need to operate the system.
- Security administrator authority (*SECADM) to users who need to create, change, or delete user profiles.
- All object authority (*ALLOBJ) to users who need to work with system resources.
- Service authority (*SERVICE) to users who need to perform service functions.
- Spool control authority (*SPLCTL) to users who need to perform all spool-related functions.
- Audit authority (*AUDIT) to users who need to perform auditing functions.

Restrictions:

1. The user profile creating or changing another user profile must have all of the special authorities being given. All special authorities are needed to grant all special authorities to another user profile.
2. A user must have *ALLOBJ and *SECADM special authorities to grant a user *SECADM special authority when using the CHGUSRPRF command.
3. The user must have *ALLOBJ, *SECADM, and *AUDIT special authorities to grant a user *AUDIT special authority when using the CHGUSRPRF command.

Specify one of the following values:

***SAME:** The value does not change.

***USRCLS:** Special authorities are given to the user based on the value entered in the USRCLS parameter.

***NONE:** No special authority is given to the user.

Or specify one or more of the following:

***SAVSYS:** Save system authority is given to the user profile. This user is given the authority to save, restore, and free storage for all objects on the system, with or without object management authority.

***JOBCTL:** Job control authority is given to the user. The user is given the authority to change, display, hold, release, cancel, and clear all jobs that are running on the system or that are on a job queue or output queue that has OPRCTL (*YES) specified. The user also has the authority to load the system, to start writers, and to stop active subsystems.

***ALLOBJ:** All object authority is given to the user. The user can access any system resource with or without private user authorizations.

***SECADM:** Security administrator authority is given to the user. The user can create, change, or delete user profiles if authorized to the Create User Profile, Change User Profile, or Delete User Profile commands and is authorized to the user profile. This right does not allow giving special authorities not in this user's profile.

***SERVICE:** Service authority is given to the user. The user can perform service functions.

***SPLCTL:** Spool control authority is given to the user. The user can perform all spool functions.

***AUDIT:** Audit authority is granted to the user. This user is given the authority to perform auditing functions. Auditing functions include turning auditing on or off for the system and controlling the level of auditing on an object or user.

PWDEXP

Specifies that the user's password is set to expired. If the password is set to expired, the user is required to change the password to sign on the system. The Sign-On Information display is shown, and the user can call the Change Password (CHGPWD) command from this display.

***SAME:** The value does not change.

***NO:** The password is not set to expired.

***YES:** The password is set to expired.

STATUS

Specifies whether the profile is in an enabled or disabled status.

***SAME:** The value does not change.

***ENABLED:** The profile is enabled.

***DISABLED:** The profile is disabled.

USRCLS

Specifies the type of user associated with this user profile: security officer, security administrator, programmer, system operator, or user. The user class determines the options shown on a menu for the user. The special authorities defined by the user class are used only if SPCAUT(*USRCLS) is specified.

***SAME:** The value does not change.

***SECOFR:** At all levels of system security, the user is given all object authority, security administrator authority, save system authority, job control authority, service authority, spool control authority, and audit authority.

***SECADM:** If the system security level is 10 or 20, the user is given all object authority, security administrator authority, save system authority, and job control authority.

If the system security level is 30 or above, the user is given security administrator authority, save system authority, and job control authority.

***PGMR:** If the system security level is 10 or 20, the user is given all object authority, save system authority, and job control authority.

If the system security level is 30 or above, the user is given save system authority and job control authority.

***SYSOPR:** If the system security level is 10 or 20, the user is given all object authority, save system authority, and job control authority.

If the system security level is 30 or above, the user is given save system authority and job control authority.

***USER:** At system security level 10 or 20, the user is granted all object and save system special authorities. At system security level 30 or above, the user is granted no special authorities.

ASTLVL

Specifies which user interface is used.

***SAME:** The value does not change.

***SYSVAL:** The system determines the graphic character set and code page values for the command parameters from the QCHRID system values.

***BASIC:** The Operational Assistant user interface is used.

***INTERMED:** The system interface is used.

***ADVANCED:** The expert system interface is used. To allow for more list entries, the options keys and the function keys are not displayed. If a command does not have an advanced (*ADVANCED) level, the intermediate (*INTERMED) level is used.

SPCENV

Specifies the special environment in which the user operates after signing on the system.

***SAME:** The value does not change.

***SYSVAL:** The system value, QSPCENV, is used to determine the system environment after the user signs on the system.

***NONE:** The user operates in the AS/400 system environment after signing on the system.

***S36:** The user operates in the System/36 environment after signing on the system.

DSPSGNINF

Specifies whether the sign-on information display is shown when the user signs on the system. This allows users to see the sign-on information, such as date of last sign-on and sign-on attempts that were not valid. If the password is due to expire in 7 days or less, the number of days until the password expires is shown.

***SAME:** The value does not change.

***SYSVAL:** The system value QDSPSGNINF is used to determine whether the sign-on information display is shown when the user signs on the system.

***NO:** The sign-on information display is not shown when the user signs on the system.

***YES:** The sign-on information display is shown when the user signs on the system.

PWDEXPITV

Specifies the number of days (from the password changed date) before a password expires.

***SAME:** The value does not change.

***SYSVAL:** The system value QPWDEXPITV is used to determine the password expiration interval.

***NOMAX:** There is no password expiration interval.

password-expiration-interval: Specify the number of days after which the password expires. Valid values range from 1 to 366.

LMTDEVSSN

Specifies whether the number of device sessions allowed for the user is limited to one. This does not limit the System Request menu or a second sign-on.

***SAME:** The value does not change.

***SYSVAL:** The system value QLMTDEVSSN is used to determine whether the user is limited to one device session.

***NO:** The user is not limited to one device session.

***YES:** The user is limited to one device session.

KBDBUF

Specifies the keyboard buffering value used when a job is initialized for this user profile. The new value takes effect the next time the user signs on. If the type-ahead feature is active, the keystrokes can be buffered. If the attention key buffering option is active, the attention key is buffered like any other key. If the attention key is not active, the attention key is not buffered and is sent to the system even if the display station is input inhibited. The keyboard buffer value can also be set by a user application using the QWSSETWS program. More information is in the *System Programmer's Interface Reference*.

***SAME:** The value does not change.

***SYSVAL:** The system value, QKBDBUF, is used to determine the keyboard buffering value for this profile.

***NO:** The type-ahead feature and attention key buffering option are not active for this user profile.

***TYPEAHEAD:** The type-ahead feature is active for this user profile.

***YES:** The type-ahead feature and attention key buffering option are active for this user profile.

MAXSTG

Specifies the maximum amount of auxiliary storage (kilobytes) assigned to store permanent objects owned by this user-described profile (1 kilobyte equals 1024 bytes). If the maximum is exceeded when an interactive user tries to create an object, an error message is displayed, and the object is not created. If the maximum is exceeded when an object is created in a batch job, an error message is sent to the job log (depending on the logging level of the job), and the object is not created.

When planning maximum storage for user profiles, consider the following system actions:

- A restore operation assigns the storage to the user doing the restore, and then transfers the object to the owner. For a large restore, specify MAXSTG(*NOMAX).
- The user profile that creates a journal receiver is assigned the required storage as the receiver size grows. If new receivers are created using JRNRCV(*GEN), the storage continues to be assigned to the user profile that owns the active journal receiver. If a very active journal receiver is owned, specify MAXSTG(*NOMAX).
- User profiles that transfer created objects to their group profile must have adequate storage in the user profiles to contain created objects before the objects are transferred to the group profile.
- The owner of the library is assigned the storage for the descriptions of objects which are stored in a library, even when the objects are owned by another user profile. Examples of such objects are text and program references.
- Storage is assigned to the user profile for temporary objects used while a job is running. Examples of such objects are commit control blocks, file editing space, and documents.

***SAME:** The value does not change.

***NOMAX:** As much storage as required is assigned to this profile.

maximum-K-bytes: Specify the maximum amount of storage (in kilobytes) that can be assigned to the user with this profile (1 kilobyte equals 1024 bytes). The maximum storage allowed can be less than the current maximum storage used for the user profile.

PTYLMT

Specifies the highest scheduling priority the user is allowed to have for each job submitted to the system. This value controls the job processing priority and output priority for any job running under this user profile; that is, values specified in the JOBPTY and OUTPTY parameters of any job command cannot exceed the PTYLMT value of the user profile under which the job is run. The scheduling priority can have a value ranging from 1 through 9, where 1 is the highest priority and 9 is the lowest priority. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

priority-limit: Specify a value ranging from 1 to 9 for the highest scheduling priority allowed to the user.

CURLIB

Specifies the name of the library being used as the current library during the processing of this command.

Note: If *PARTIAL or *YES is specified for the LMTCPB parameter in the user profile, the user cannot change the current library at sign-on or by using the Change Profile (CHGPRF) command.

***SAME:** The value does not change.

***CRTDFT:** This user has no current library. If objects are created and placed in the current library by using *CURLIB on a create command, the QGPL library is used as the default current library.

current-library-name: Specify the name of the library that is this user's current library after sign-on to the system.

INLPGM

Specifies, for an interactive job, the name of the program called whenever a new routing step is started that has QCMD as the request processing program. If *PARTIAL or *YES is specified on the LMTCPB parameter in the user profile, the program value cannot be changed at sign on or by using the Change Profile (CHGPRF) command. No parameters can be passed to the program.

A system/36 environment procedure name can be specified as the initial program if the procedure is a member of the file QS36PRC (in the library list or specified library) and if either of the following conditions are true:

- *S36 is specified for the SPCENV parameter.
- *SYSVAL is specified for the SPCENV parameter and the system value, QSPCENV, is *S36.

***SAME:** The value does not change.

***NONE:** No program is called when the user signs on the system. If a menu name is specified in the INLMNU parameter, that menu is displayed.

The name of the program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the program called after the user signs on the system.

INLMNU

Specifies the name of the initial menu displayed when the user signs on the system if the user's routing program is the command processor QCMD. If *YES is specified for the LMTCPB parameter in the user profile, the user cannot change the menu either at sign-on or with the Change Profile (CHGPRF) command.

A system/36 environment menu can be specified as the initial menu if either of the following conditions are true:

- *S36 is specified for the SPCENV parameter.

- *SYSVAL is specified for the SPCENV parameter and the system value, QSPCENV, is *S36.

***SAME:** The value does not change.

***SIGNOFF:** The system signs-off the user when the initial program completes. This is intended for users authorized only to run the program.

The name of the menu can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

menu-name: Specify the name of the initial menu called after the user signs on the system.

LMTCPB

Specifies the limit to which the user can control the program, menu, current library, and the ATTN key handling program values. It also determines whether the user can run commands from a command line. This parameter is ignored when the security level is 10.

Note: When creating or changing other users' user profiles, users running this command cannot specify values on this parameter that grant greater capabilities to other users than their own user profiles grant to them. For example, if *PARTIAL is specified on the LMTCPB parameter in the user profile of the user running this command, *PARTIAL or *YES can be specified for another user. *NO cannot be specified for another user.

***SAME:** The value does not change.

***NO:** The program, menu, and current library values can be changed when the user signs on the system. Users may change the program, menu, current library, or ATTN key handling program values in their own user profiles with the Change Profile (CHGPRF) command. Commands can be run from a command line.

***PARTIAL:** The program and current library cannot be changed on the sign-on display. The menu can be changed and commands can be run from a command line. A user can change the menu value with the Change Profile (CHGPRF) command. The program, current library, and the ATTN key handling program cannot be changed using the Change Profile command.

***YES:** The program, menu, and current library values cannot be changed on the sign-on display. Commands cannot be run from a command line, but can still be run from a command entry screen. The user cannot change the program, menu, current library, or the ATTN key program handling values by using the Change Profile (CHGPRF) command.

JOBD

Specifies the name of the job description used. If the job description does not exist when the entry is changed, a library qualifier must be specified because the qualified job description name is retained in the subsystem description. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The job description name does not change.

The name of the job description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

job-description-name: Specify the name of the job description used for the work station entries whose job description parameter values indicate the user (JOBD(*USRPRF)).

GRPPRF

Specifies the user's group profile name whose authority is used if no specific authority is given for the user. The current user of this command must have *OBJMGT and *CHANGE authority to the profile specified on the GRPPRF parameter. The required *OBJMGT authority cannot be given by a program adopt operation.

Notes:

1. When a group profile is specified, the user is automatically granted *CHANGE and *OBJMGT authority to the group profile.
2. The following IBM-supplied objects are not valid on this parameter:

QDBSHR	QRJE
QDFTOWN	QSNADS
QDOC	QSPL
QDSNX	QSPLJOB
QFNC	QSRV
QGATE	QSRVBAS
QLPAUTO	QSYS
QLPINSTALL	QTSTRQS

***SAME:** The value does not change.

***NONE:** This user profile has no group profile.

user-profile-name: Specify the name of the group profile used with this user profile.

OWNER

Specifies the user profile that is the owner of newly created objects.

***SAME:** The value does not change.

***USRPRF:** The user profile associated with the job is the owner of the object.

***GRPPRF:** The group profile is made the owner of newly created objects and is given all authority to the object. The user profile associated with the job does not have any specific authority to the object. If *GRPPRF is specified, a user profile name must be in the GRPPRF parameter, and the GRPAUT parameter cannot be specified.

GRPAUT

Specifies the specific authority given to the group profile for newly created objects. If *GRPPRF is specified on the OWNER parameter, specification of this parameter is not allowed.

***SAME:** The value does not change.

***CHANGE:** The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority.

***ALL:** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence and specify the security for the object, change the object, and perform basic functions on the object. The user can change ownership of the object.

***USE:** The user can perform basic operations on the object, such as running a program or reading a file. The user cannot change the object. *USE authority provides object operational authority and read authority.

***EXCLUDE:** The user cannot access the object.

ACGCDE

Specifies the accounting code associated with this user profile. More information on job accounting is in the *Database Guide*.

***SAME:** The value does not change.

***BLANK:** An accounting code consisting of 15 blanks is assigned to this user profile.

accounting-code: Specify the 15-character accounting code to be used by jobs that get their accounting code from this user profile. If less than 15 characters are specified, the string is padded on the right with blanks.

DOCPWD

Specifies the document password that allows Document Interchange Architecture (DIA) users to protect personal distributions from being used by users working on their behalf. More information on specifying a document password is in the *Managing OfficeVision/400** manual.

***SAME:** The value does not change.

***NONE:** No document password is used by this user.

document-password: Specify a document password for use by this user. The password must consist of alphanumeric characters ranging from 1 through 8 (letters A through Z and numbers 0 through 9). The first character of the document password must be alphabetic; the remaining characters can be alphanumeric. Embedded blanks, leading blanks, and special characters are not valid.

MSGQ

Specifies the qualified name of the message queue to which messages are sent.

Note: The message queue is created if it does not already exist. The user profile being changed is the owner of the message queue.

***SAME:** The value does not change.

***USRPRF:** A message queue with the same name as that specified in the USRPRF parameter is used as the message queue for this user. QUSRSYS is the library where this message queue is located.

The name of the message queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

message-queue-name: Specify the name of the message queue used by this user.

DLVRY

Specifies how the messages sent to the message queue for the user profile are delivered.

***SAME:** The value does not change.

***HOLD:** The messages are held in the message queue until they are requested by the user or program.

***BREAK:** The job, to which the message queue is assigned, is interrupted when a message reaches the message queue. If the job is an interactive job, the audible alarm is sounded (if the alarm is set). The delivery mode cannot be changed to *BREAK if the message queue is also being used by another job.

***NOTIFY:** The job to which the message queue is assigned is notified when a message reaches the message queue. For interactive jobs at a work station, the audible alarm is sounded (if the alarm is set) and the message-waiting light is turned on. The delivery mode cannot be changed to *NOTIFY if the message queue is also being used by another job.

***DFT:** The default reply to the inquiry message is sent. If no default reply is specified in the message description of the inquiry message, the system default reply, *N, is used.

SEV

Specifies the lowest severity code that a message can have and still be delivered to a user in break or notify

mode. Messages arriving at the message queue whose severities are lower than the severity code specified on this parameter do not interrupt the job or turn on the audible alarm or the message-waiting light; they are held in the queue until they are requested by using the Display Message (DSPMSG) command. If *BREAK or *NOTIFY is specified on the DLVRY parameter, and is in effect when a message arrives at the queue, the message is delivered if the severity code associated with the message is equal to or greater than the value specified here. Otherwise, the message is held in the queue until it is requested.

***SAME:** The value does not change.

severity-code: Specify severity code ranging from 00 through 99.

PRTDEV

Specifies the qualified name of the default printer device for this job. If OUTQ(*DEV) is specified, the file is placed on an output queue with the same name as the printer.

***SAME:** The value does not change.

***WRKSTN:** The printer assigned to the user's work station is used.

***SYSVAL:** The value specified in the system value QPRTDEV is used.

print-device-name: Specify the name of a printer used to print the output for this user.

OUTQ

Specifies the qualified name of the output queue.

***SAME:** The value does not change.

***WRKSTN:** The output queue assigned to the user's work station is used.

***DEV:** An output queue with the same name as that specified on the *DEV parameter of the Create Printer File (CRTPRTF), Change Printer File (CHGPRTF), or Override with Printer File (OVRPRTF) command is used as the output queue for this user.

The name of the output queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

output-queue-name: Specify the name of the output queue used as the output queue for this user.

ATNPGM

Specifies the program to be the ATTN key handling program for this user. The ATTN key handling program is called when the ATTN key is pressed during an interactive job. The program is active only when the user specifies routes to the system-supplied QCMD command processor. The ATTN key handling program is activated before the program (if any) is called and is active for both program and menu. If the program changes the ATNPGM (by using the Set Attention Program (SETATNPGM) command), the new program remains active only for the duration of the program. When control returns and QCMD calls the menu, the original ATTN key handling program becomes active again. If the SETATNPGM command is run from the menus or an application is called from the menus, the new ATTN key handling program specified overrides the original ATTN key handling program. If *YES or *PARTIAL is specified on the LMTCPB parameter in the user profile, the ATTN key handling program cannot be changed by the Change Profile (CHGPRF) command.

***SAME:** The value does not change.

***ASSIST:** QEZMAIN is used.

***SYSVAL:** The system value QATNPGM is used.

***NONE:** No ATTN key handling program is used by this user.

The name of the ATTN key handling program can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the ATTN key handling program used by this user.

SRTSEQ

Specifies the sort sequence table to be used for string comparisons for this user profile.

***SAME:** This value does not change.

***SYSVAL:** The system value QSRTSEQ is used.

***HEX:** A sort sequence table is not used. The hexadecimal values of the characters are used to determine the sort sequence.

***LANGIDUNQ:** A unique-weight sort table is used.

***LANGIDSHR:** A shared-weight sort table is used.

The name of the sort sequence table can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

table-name: Specify the name of the sort sequence table to be used with this user profile.

LANGID
Specifies the language identifier used for this user.

***SAME:** The language identifier does not change.

***SYSVAL:** The system value QLANGID is used.

language-identifier: Specify the language identifier. More information on valid language identifiers in the *National Language Support Planning Guide*.

CNTRYID
Specifies the country identifier to be used by the system for this user.

***SAME:** The country identifier does not change.

***SYSVAL:** The system value QCNTRYID is used.

country-identifier: Specify an ISO 3166 Alpha-2 code from the country code table. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

CCSID
Specifies the coded character set identifier (CCSID) used for this user.

A CCSID is a 16-bit number identifying a specific set of encoding scheme identifiers, character set identifiers, code page identifiers, and additional coding-related information that uniquely identifies the coded graphic representation used.

Note: If the value for CCSID is changed, the change does not affect jobs that are currently running.

***SAME:** The CCSID does not change.

***SYSVAL:** The system value QCCSID is used.

***HEX:** The CCSID 65535 is used.

coded-character-set-identifier: Specify the CCSID. More information on valid CCSIDs is in the *National Language Support Planning Guide*.

USROPT
Specifies the level of information detail the user can view and the function of the Page Up and Page Down keys by default. The system shows displays suitable for the inexperienced user. More experienced users must perform an extra action to see more detailed information. When values are specified for this parameter, the system presents detailed information without additional action by the user.

***SAME:** The value does not change.

***NONE:** No detailed user option information is shown.

***CLKWD:** Parameter keywords are shown instead of the possible parameter values when a control language (CL) command is displayed.

***EXPERT:** More detailed user option information is initially shown when the user is performing display and edit options (such as edit or display object authority) to define or change the system.

***ROLLKEY:** The actions of the Page Up and Page Down keys are reversed.

***HLPFULL:** Help text is shown on a full display rather than in a window.

***NOSTSMSG:** Status messages are not displayed when sent to the user.

***STSMSG:** Status messages are displayed when sent to the user.

***PRTMSG:** A message is sent to this user's message queue when a spooled file for this user is printed or held by the printer writer.

TEXT
Specifies text that briefly describes the user ID named in the USRPRF parameter. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

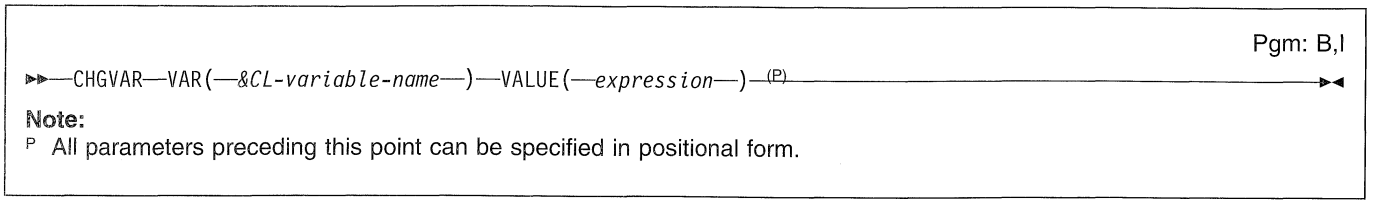
```
CHGUSRPRF  USRPRF(JJADAMS)  PASSWORD(SECRET)
  SPCAUT(*JOBCTL)  INLPGM(ARLIB/DSPMENU)
```

This command makes the following changes to the user profile named JJADAMS:

- Changes the password to SECRET.
- Authorizes JJADAMS to use the special job control authority.
- Changes the first program to start following a successful sign-on to a program named DSPMENU, which is located in a library named ARLIB.

All the other command parameters default to *SAME and do not change.

CHGVAR (Change Variable) Command



Purpose

The Change Variable (CHGVAR) command is used in CL programs to change the value of a CL variable or to change part of a character variable by using the substring built-in function (%SUBSTRING) or the binary built-in function (%BINARY). The value can be changed to the value of a constant, to the value of another variable, or to the value gotten from the evaluation of an expression or a built-in function. Expressions and built-in functions are described in Chapter 3, "Expressions in CL Commands." Also, implicit conversion between decimal and character values is performed by the rules given in the VALUE parameter description.

The binary built-in function (%BINARY or %BIN) can be used in either the VAR or the VALUE parameter as a substitute for a decimal variable. When used with the VAR parameter, the specified portion of the character variable is changed to the signed binary integer equivalent value of the arithmetic expression given in the VALUE parameter. When used within the VALUE parameter, the specified portion of the character variable is treated as a signed binary integer converted to a decimal number when used in evaluating the value of the VALUE parameter. A 2-byte binary integer is converted to a decimal (5 0) number and a 4-byte binary number is converted to decimal (10 0) number. The result of the evaluated expression is then assigned to the specified in the VAR parameter.

The substring built-in function (%SUBSTRING or %SST) can be used in either the VAR or the VALUE parameter as a substitute for a character variable. When used with the VAR parameter, the specified portion of the character variable is changed to the value of the expression given in the VALUE parameter. When used within the VALUE parameter, the specified portion of the character variable is used in evaluating the value of the VALUE parameter. 2-byte binary integers are converted to a decimal (50)numbers and a 4-byte binary numbers are converted to decimal (10 0) numbers. The result of the evaluated expression is then assigned to the specified in the VAR parameter.

The substring built-in function can be used to retrieve or change all or part of the local data area associated with a job.

The %SWITCH built-in function can be used in the VALUE parameter as a substitute for a logical variable declared in the program. %SWITCH contains an 8-character mask that indicates which of the eight job switches in a job are tested for 1s and 0s. When %SWITCH is specified for the VALUE parameter, the logical variable specified by the VAR parameter is set to '1' if the logical results of the built-in function are all true. If any of the job switches tested results in a false condition, the variable is set to '0'.

Restriction: The CHGVAR command is valid only in CL programs.

Required Parameters

VAR

Specifies the name of the CL variable whose value is being changed. The type of variable does not have to be the same as the type of constant or variable specified in the VALUE parameter, unless an expression is being evaluated or the VAR parameter specifies a logical variable.

If the substring built-in function or the binary built-in function is used to change a portion of a character variable (that is, a substring of the character string in the variable) specified in VAR to a value specified in the VALUE parameter, specify the name of the character variable, followed by the starting position and the number of characters being changed within the character string specified by the variable name.

VALUE

Specifies the expression that is used to change the value of the variable. Variables, constants, or a built-in function can be used within the expression. For a description of expressions, see Chapter 3, "Expressions in CL Commands."

If a constant is used as a simple expression, its value must be specified by the following rules, depending on the type of constant being specified and whether the variable was declared as a decimal, character, or logical variable.

Coding Decimal Values for Decimal Variables

When a numeric value is specified for a decimal variable:

- It can be coded with or without a decimal point (. or ,) and with or without a plus or minus sign.
- If a negative value is specified, a minus sign (–) must precede the value.

- If a decimal point is not specified in the coded value, it is assumed to be on the right of the last digit specified; that is, the coded value is assumed to be an integer (whole number).
- If the number of either integer or fractional digits specified is greater than the defined number of integer or fractional digits, an error message is sent to the user.

For example, if a decimal variable is defined as a five-position decimal value of which two positions are the fraction portion, the following values can be coded:

Specified Value	Assumed Value
2.7 or 2,7	2.70
27 or 27.00	27.00
-27	-27.00

Coding Character Values for Decimal Variables

When a character value is specified for a decimal variable:

- Only the digits 0 through 9, a decimal point (. or ,), and a plus sign (+) or minus sign (–) can be used.
- If a plus sign or minus sign is specified, it must be placed immediately in front of (no blanks between) the first digit in the character value. If no sign character is specified, the value is converted as a positive value.
- The number of decimal positions in the converted result is determined by the decimal point specified in the character value. If no decimal point is specified, it is assumed to be to the right of the last digit in the converted value.
- Decimal alignment occurs in the converted result. The number of decimal positions in the converted result is determined by the number declared for the variable. If the specified character value has more decimal positions than the declared variable, the extra positions on the right are truncated. If the integer portion of the character value has more digits than that declared for the variable, an error message is sent to the user.

The following examples show the results of converting the indicated character values for character variable &A to decimal values for decimal variable &B.

```
CHGVAR    VAR(&B)  VALUE(&A)
```

Character Variable &A		Decimal Variable &B	
Length	**Specified Value**	**Length**	**Converted Result**
10	'+123.1'	5, 2	123.10
10	'+123.00'	5, 0	123
10	'-123'	5, 2	-123.00

When the binary built-in function is used instead of the decimal variable &B, the decimal value is converted to a signed binary number.

Coding Character Values for Character Variables

When a character string is specified for a character variable, it must be enclosed in apostrophes if it contains special characters or consists entirely of numeric characters. For example, 'ABC 67', which contains a blank, or '37.92', which contains a decimal point and consists entirely of numeric characters. If 37.92 is not enclosed in apostrophes, it is handled as a decimal value instead of a character value.

Character variables are padded with blanks (or are truncated) on the right if the character string for the VALUE parameter is shorter (or longer) than the variable specified by the VAR parameter.

If a character variable is set equal to a portion of another character variable, specify, as parameters on the substring built-in function, the name of the variable containing the substring, the starting character position, and the number of characters being replaced. The starting position and the number of characters can be specified in CL variables.

Coding Decimal Values for Character Variables

When a decimal value is specified for a character variable:

- The same digits, decimal point, and sign character (if the value is negative) are used in the converted result. The value is right-justified in the character variable and padded on the left with zeros, if needed (this is unique to converted CL decimal values).
- The converted result has as many decimal positions as were specified in the decimal value or as defined for the decimal variable being used. If no decimal positions are specified in the decimal value or defined for the decimal variable, no decimal point is placed in the result.
- A minus sign is placed in the leftmost position of the character variable if the specified decimal value is negative. No plus sign is placed in the character variable for positive values.

The following examples show the results of converting the indicated decimal values for decimal variable &B to character values for character variable &A.

```
CHGVAR    VAR(&A)  VALUE(&B)
```

When the binary built-in function is used instead of the decimal variable &B, the signed binary number is converted to a decimal number.

Decimal Variable &B		Character Variable &A	
Length	**Specified Value**	**Length**	**Converted Result**
5, 2	23.00 or +23	7	0023.00
5, 2	-3.9	7	-003.90
5, 2	-123.67	7	-123.67

Note: The character variable must be long enough to accommodate the decimal point and sign character if the value can have a decimal point and a negative value in it. In the last example, although the decimal

value is defined as (5, 2), the character variable must be at least 7 characters long for the value shown. In the next-to-last example, the character variable could only be 5 characters long and the converted result -3.90 would be valid.

The substring built-in function can be used to change a substring of a character variable specified in the VAR parameter to a decimal value in the VALUE parameter.

Coding Logical or Character Values for Logical Variables. The value for a logical variable must be a logical value of either '1' or '0'. It must be enclosed in apostrophes. However, the %SWITCH built-in function can be used in place of a logical variable in the VALUE parameter. Refer to Chapter 3, "Expressions in CL Commands" for a description of the %SWITCH built-in function.

Note: Values for decimal and character variable types can be specified in hexadecimal form (X'580F' for decimal 58.0). However, if character values are specified in hexadecimal form, care should be used because no validity checking is performed on the hexadecimal string.

The following examples of the CHGVAR command show how the values of decimal, logical, and character variables can be changed.

Examples

Example 1: Changing Decimal Variables

```
CHGVAR   &A  &B
```

The value of variable &A is set to the value of the variable &B. If &B has a value of 37.2, then the value of &A becomes 37.2 also.

```
CHGVAR   &Y  (&Y+ 1)
```

The value of variable &Y is increased by 1. If &Y has a value of 216, its value is changed to 217.

Example 2: Changing Logical Variables

```
CHGVAR   &X  (&Y *OR &Z)
```

The value of the logical variable &X is set to the value of the result of the OR operation of the logical variable &Y with the logical variable &Z. Both variables *must* be logical variables when *OR is used. If &Y equals '0' and &Z equals '1', then &X is set to '1'.

```
CHGVAR   &A  %SWITCH(10XXXX10)
```

The value of the logical variable &A is determined by the logical results of the built-in function, %SWITCH. Positions 1, 2, 7, and 8 of the 8-character mask indicate that the corresponding job switches for the job are to be tested for the values indicated in the mask. Job switches 1 and 7 are tested for 1s, and switches 2 and 8 are tested for 0s. (Switches 3 through 6 are not tested.) If all four switches contain the values specified in the %SWITCH mask, the logical result of the built-in function is true, and the variable &A is set to a '1'. If any of the four switches contain a value not indicated in the mask, the result is false and &A is set to '0'.

Example 3: Changing Character Variables

```
CHGVAR   VAR(&A)  VALUE(AB *CAT CD)
  CHGVAR   &A  ('AB' *CAT 'CD')
```

These two commands set the value of the variable &A equal to the character string ABCD, which is the result of the concatenation of the two character strings AB and CD. The first command is coded in keyword form with unquoted strings; the second command is coded in positional form with the VALUE parameter specifying two quoted character strings.

```
CHGVAR   &VAR1  &VAR2
```

This example shows a 6-character variable whose value is changed by a shorter character string. If &VAR1 = ABCDEF and &VAR2 = XYZ before the command is processed, the result in &VAR1 is padded on the right with blanks: XYZbbb.

```
CHGVAR   &VAR1  '12'
```

Assuming &VAR1 is a character variable that is 6 characters long, the result is again padded on the right with blanks: 12bbbb. The apostrophes are required in this example.

```
CHGVAR   VAR(%SUBSTRING(&A 4 3))  VALUE(REP)
          or
CHGVAR   VAR(%SST(&A 4 3))  VALUE(REP)
```

The substring built-in function is used to change 3 characters of the character constant in the variable named &A. If &A has a value of ABCDEFGH, the fourth, fifth, and sixth characters in &A are set to REP, and the result is ABCREPGH.

```
CHGVAR   VAR(%SST(*LDA 1 512))  VALUE(' ')
```

The substring built-in function is used to change all of the local data area to blanks.

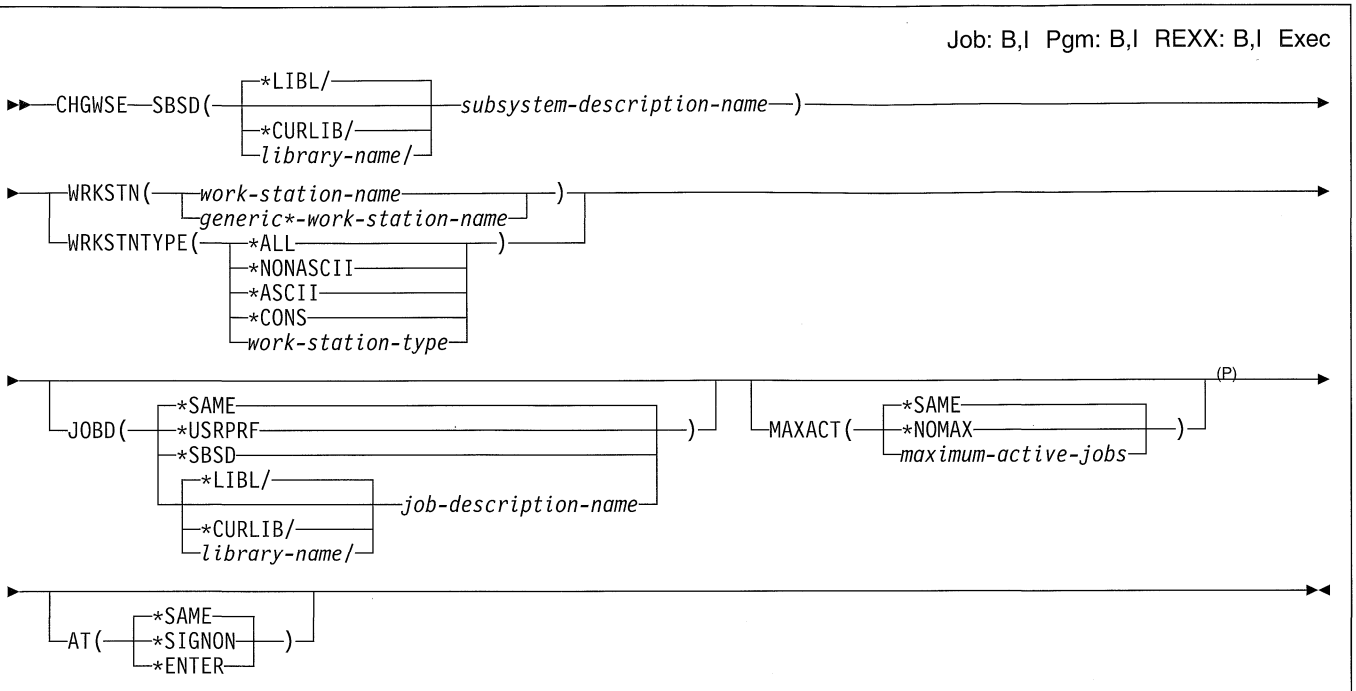
```
  CHGVAR  VAR(%BINARY(&A 1 2))  VALUE(20)
```

or

```
  CHGVAR  VAR(%BIN(&A 1 2))  VALUE(20)
```

The binary built-in function is used to change the first 2 characters of the character variable named &A to the signed binary value of the number 20, or hexadecimal number X'0014'. If the character variable named &A has a length of 10, characters 3 through 10 of variable &A are not changed.

CHGWSE (Change Work Station Entry) Command



Note:

| P All parameters preceding this point can be specified in positional form.

Purpose

The Change Work Station Entry (CHGWSE) command changes one or more attributes of a work station entry in the specified subsystem description. The associated subsystem must be inactive when the changes are made.

Restriction: The user of this command must have object operational and object management authorities for the subsystem description, and object operational authority for the job description.

Required Parameters

SBSD

Specifies the qualified name of the subsystem description that contains the work station entry that is changed.

| The name of the subsystem description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

subsystem-description-name: Specify the name of the subsystem description that contains the work station entry that is being changed.

WRKSTN

Specifies the device description name of the work station whose work station entry is being changed.

The WRKSTN parameter and the WRKSTNTYPE parameter are mutually exclusive.

work-station-name: Specify the name of the work station whose work station entry is being changed.

generic-work-station-name:* Specify the generic name
| of the work station. A generic name is a character string
| of one or more characters followed by an asterisk (*); for
| example, ABC*. The asterisk substitutes for any valid
| characters. A generic name specifies all objects with
| names that begin with the generic prefix for which the
| user has authority. If an asterisk is not included with the
| generic (prefix) name, the system assumes it to be the
| complete object name. If the complete object name is
| specified, and multiple libraries are searched, multiple
| objects can be changed only if *ALL or *ALLUSR library
| values can be specified for the name. For more infor-
| mation on the use of generic functions, refer to "Rules
| for Specifying Names."

WRKSTNTYPE

Specifies the type of work station whose work station entry is being changed. This work entry applies to all

work stations of this type that do not have specific work entries for an individual work station.

The WRKSTN parameter and the WRKSTNTYPE parameter are mutually exclusive.

The following type codes are valid:

Type Code	Device
3179	3179 Display Station
3180	3180 Display Station
3196	3196 Display Station
3197	3197 Display Station
3277	3277 Display Station
3278	3278 Display Station
3279	3279 Display Station
3476	3476 Display Station
3477	3477 Display Station
3486	3486 Display Station
3487	3487 Display Station
5251	5251 Display Station
5291	5291 Display Station
5292	5292 Color Display Station
5555	5555 Display Station (on systems supporting DBCS (double-byte character set))

***ALL:** The work station entry for all valid work station types is added.

***NONASCII:** The work station entry for all valid work stations that use 5250 data streams is added.

***ASCII:** The work station entries for all work stations that use ASCII data streams are added.

***CONS.:** This value overrides a device type entry that specifies the same device type as the device being use as the console.

work-station-type: Specify the work station device type whose work station entry is being changed.

Optional Parameters

JOBD

Specifies the name of the job description used. If the job description does not exist when the entry is changed, a library qualifier must be specified because the qualified job description name is retained in the subsystem description. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***USRPRF:** The job description named in the user profile of the user that signs on at this work station (or at this type of work station) is used for jobs started through this entry.

***SBSD:** The job description that has the same name as the subsystem description, specified by the SBSD parameter, is used for jobs created through this entry.

The name of the job description can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

job-description-name: Specify the qualified name of the job description that is used for jobs created through this entry.

MAXACT

Specifies, for work stations that use this work station entry, the maximum number of work station jobs that can be active (or signed on) at the same time. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***SAME:** The value does not change.

***NOMAX:** There is no maximum number of jobs that can be active at the same time.

maximum-active-jobs: Specify the new maximum number of jobs that can be active at the same time through this entry.

AT Specifies when the work stations associated with this work entry are allocated. For more information on how work stations are allocated to subsystems, see the Start Subsystem (STRSBS) command description.

Note: If two or more work station entries specify AT(*SIGNON) for the same work station, if they are in more than one subsystem description, and if the work station is varied on while more than one of the subsystems are active, it cannot be predicted to which subsystem the work station will be assigned.

***SAME:** The value does not change.

***SIGNON:** The work stations are allocated when the subsystem is started. A sign-on prompt is shown at each work station associated with this work entry. If a work station becomes allocated to a different subsystem, interactive jobs associated with the work station are allowed to enter this subsystem through the Transfer Job (TFRJOB) command.

***ENTER:** The work stations associated with this work entry are not allocated when the subsystem is started. However, the interactive jobs associated with the work stations are allowed to enter this subsystem through the TFRJOB command.

Examples

Example 1: Changing an Entry at Signon

```
CHGWSE  SBSD(QGPL/BAKER)  WRKSTN(A12)  AT(*SIGNON)
```

This command changes the work station entry for work station A12 in subsystem BAKER found in the general

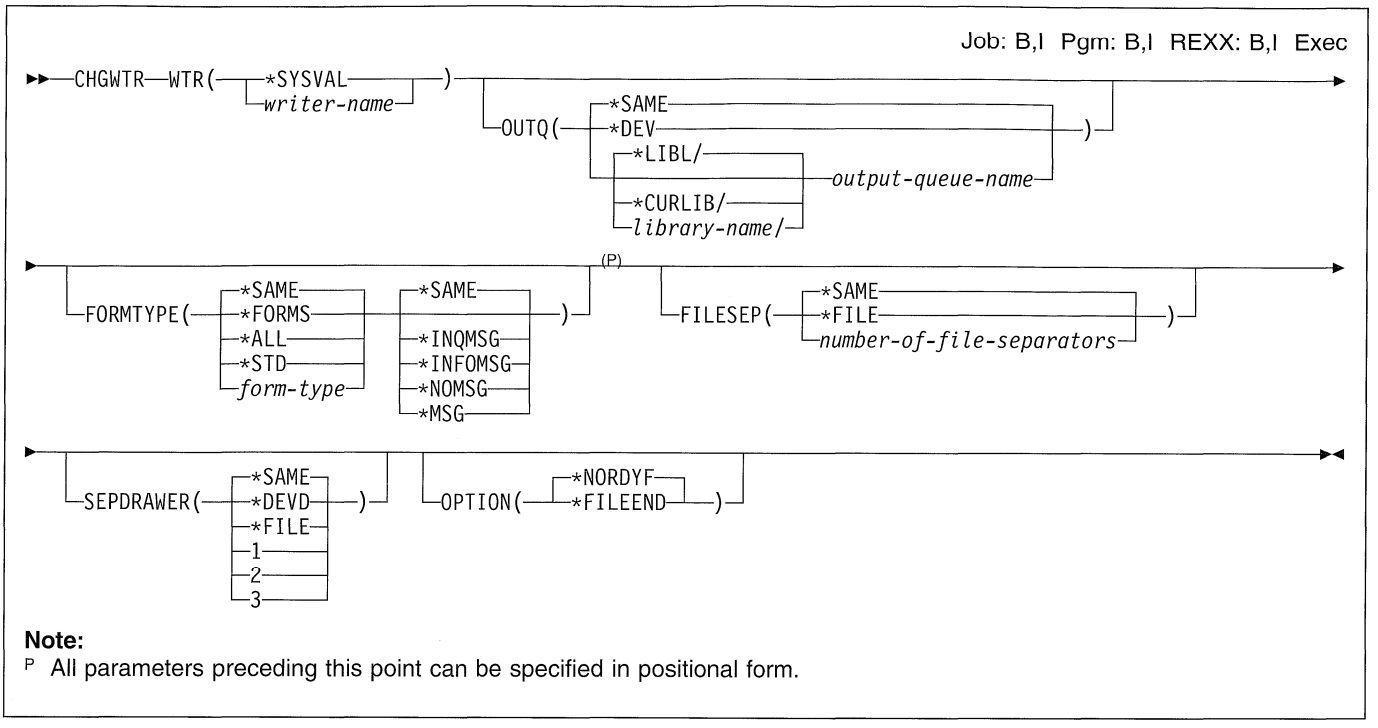
purpose library. A job is created for work station A12 when the user's password is entered on the sign-on prompt and the Enter key is pressed.

Example 2: Changing an Entry

```
CHGWSE  SBSD(QGPL/BAKER)  WRKSTN(B28)
```

This command changes the job entry for work station B28 in subsystem BAKER found in the general purpose library. The JOBD, MAXACT and AT parameters default to the *SAME value.

CHGWTR (Change Writer) Command



Note:
P All parameters preceding this point can be specified in positional form.

Purpose

The Change Writer (CHGWTR) command allows the user to change the form type, number of file separators, and output queue attributes of an active printer writer. This capability provides the best performance when you want to process all files of one form type or output queue, and then all files of another form type or output queue, instead of ending and restarting the writer or changing to a different output queue each time.

If changes are made while the writer is in hold (HLD) status, the changes do not take effect until after the writer is released. The change is then made based on the value specified on the OPTION parameter.

Required Parameter

WTR

Specifies the simple name of the printer writer being changed. Each writer name must be unique.

***SYSVAL:** The writer name of the system default printer is changed.

writer-name: Specify the name of the writer being changed.

Optional Parameters

OUTQ

Specifies the qualified name of the output queue.

***SAME:** The output queue being used is not changed.

***DEV:** Specifies the default output queue associated with the printer is used.

The name of the output queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

output-queue-name: Specify the name of the output queue.

FORMTYPE

Specifies the form type selection codes for which the spooled files are being produced by the writer. A file's form type is originally derived from the form type specified in the device file that produced the spooled file. This parameter specifies that only the files of this form type are produced by the writer. All other files are left on the output queue as available.

Note: The form load message is issued when the spooled file to be printed has a form type different from the form type of the last spooled file that was printed on the device. The last form type printed is kept from the last STRPRTWTR, CHGWTR, or VRYCFG command issued.

Consider the following example:

1. The last spooled file printed on printer PRT01 had the form type *STD.
2. The user changes the form type on PRT01 to XYZ using the following command:

   ```
   CHGWTR  PRT01  FORMTYPE(XYZ)
   ```

3. No spooled file with the form type XYZ is printed on PRT01.
4. The user then sends a spooled file with the form type *STD to PRT01. The form load message is not issued, despite the intervening CHGWTR command, because the last spooled file printed on PRT01 had the same form type as the spooled file being printed.

   The form load message would be issued if a spooled file with the form type XYZ were actually printed on PRT01.

Element 1: Type of Form Designation

***SAME:** The value does not change.

***FORMS:** All available files with the same form type are produced as a group before the writer moves on to the next form type. The writer initially chooses the first available file on the queue. After the first file is complete, all files with the same form type as the first are processed. The writer again chooses the first available file on the queue and repeats the process.

***ALL:** All form types are produced by the writer.

***STD:** Only files that specify the standard form type are selected.

form-type: Specify the type of form for which the spooled files are produced.

Element 2: Message Sending Options

***SAME:** The value does not change.

***INQMSG:** An inquiry message is sent when the current file has a form type that is different than the one loaded on the device.

***INFOMSG:** An informational message is sent when the writer runs out of files with the specified form type.

***NOMSG:** No informational message is sent when the writer runs out of files with the specified form type. Also, no inquiry message is sent when the current file has a form type that is different than the one loaded on the device.

***MSG:** The writer sends an informational message when it runs out of files with the specified form type. If the printer writer does not end, (specified by the AUTOEND parameter on the STRPRTWTR command), then as additional spooled files become available for printing they are produced. The informational message is sent each time the writer must wait for additional spooled files.

FILESEP

Specifies the number of file separator pages to print preceding each file.

***SAME:** The value does not change.

***FILE:** Print the number of file separator pages that is specified for each individual file.

number-of-file-separators: Specify the number of file separator pages to print.

SEPDRAWER

Specifies which drawer is selected for printing file and job separators.

***SAME:** The value does not change.

***DEVD:** The value stored in the device description for the printer is used.

***FILE:** The separator pages are printed from the same drawer as the spooled file.

1: The separator pages are printed from drawer 1.

2: The separator pages are printed from drawer 2.

3: The separator pages are printed from drawer 3.

OPTION

Specifies when the writer change occurs.

***NORDYF:** The writer change occurs when there are no files on the output queue that meet the writer's current form type selection requirements.

***FILEEND:** The writer change occurs at the end of the current file.

Example

```
CHGWTR  WTR(MYWTR)  FORMTYPE(MYFORM *NOMSG)
  OPTION(*FILEEND)
```

This command changes writer MYWTR, which has been producing files of some other form type, to produce files with a form type of MYFORM at the end of the file now being produced. The writer is also prevented from sending an informational message when it runs out of eligible files with form type MYFORM.

Additional Considerations

By using the CHGWTR command to control the form type being produced by a writer, the operator can improve efficiency by minimizing the number of form changes required. On the other hand, if the print writer's output queue has a large number of spooled files with the wrong form type, which must be bypassed by the writer, then writer output queue search time increases and total system performance may decrease.

It is possible to enter another change writer command for a writer before the previous change has taken place. When this is done the later change is not queued, but it updates the first change instead. For example, if the command

```
CHGWTR  WTR(WRKL)  OUTQ(QPRINT)  OPTION(*FILEEND)
```

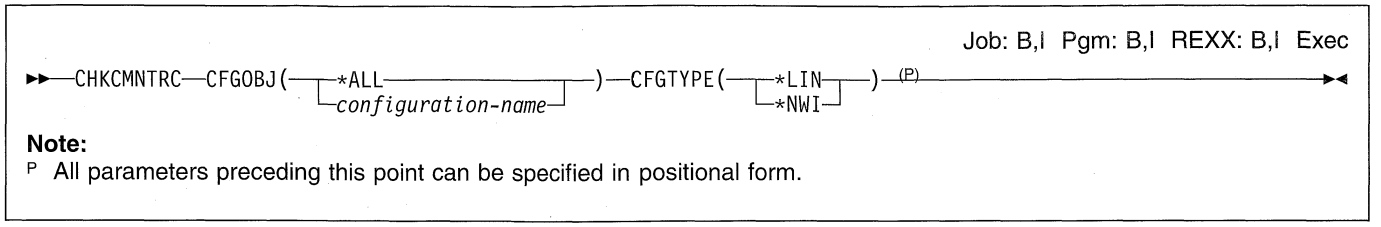
is entered, then (before the change takes place) the command

```
CHGWTR  WTR(WRKL)  FORMTYPE(XYZ *SAME)
  OPTION(*FILEEND)
```

is entered, the output queue of writer WRKL changes to QPRINT, and the form type is changed to XYZ when the writer finishes producing the current file.

Whenever the output queue or form type is changed for a writer, the print writer starts searching the output queue from the beginning and selects the first available file on the queue. This occurs even if the change takes place between two SCHEDULE(*JOBEND) files of the same job.

CHKCMNTRC (Check Communications Trace) Command



Note:

P All parameters preceding this point can be specified in positional form.

Purpose

The Check Communications Trace (CHKCMNTRC) command returns the communications trace status for a specific line or network interface description, or for all of the traces of a specific type that exist on the system. The status is returned through a message.

Restrictions:

1. You must have *SERVICE special authority to use this command.
2. The following user profiles have authority to this command:
   - QSECOFR
   - QSRV

Required Parameters

CFGOBJ

Specifies the name of the configuration description to check. The configuration description is either a line description or a network interface description.

***ALL:** The communications trace status is returned for all of the traces of a specific type.

configuration-name: Specify the configuration object name.

CFGTYPE

Specifies the object type of the configuration description to check.

***LIN:** Status for lines is shown.

***NWI:** Status for network interfaces is shown.

Examples

Example 1: Checking All Traces

```
CHKCMNTRC  CFGOBJ(*ALL)  CFGTYPE(*NWI)
```

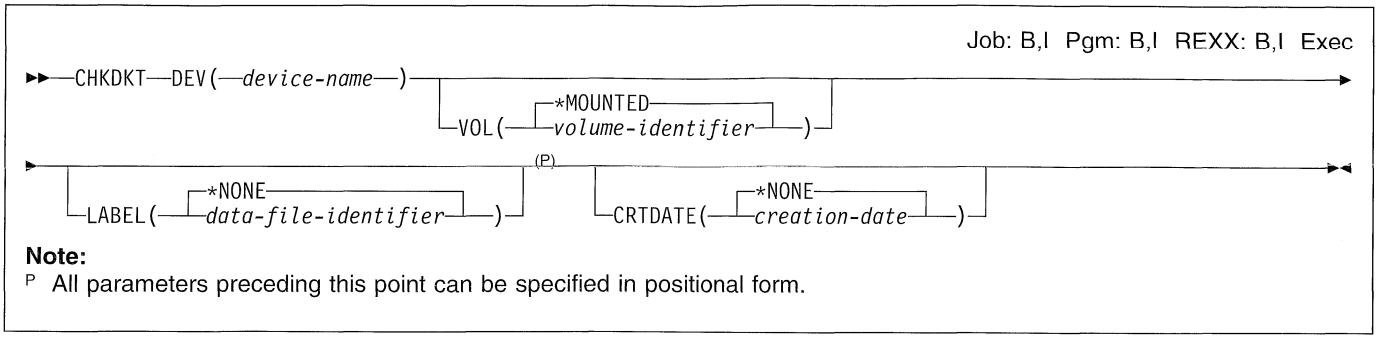
This command shows the communications trace status of all network interface traces.

Example 2: Checking An Individual Trace

```
CHKCMNTRC  CFGOBJ(*QESLINE)  CFGTYPE(*LIN)
```

This command shows the communications trace status of line QESLINE.

CHKDKT (Check Diskette) Command



Note:

| P All parameters preceding this point can be specified in positional form.

Purpose

The Check Diskette (CHKDKT) command searches a diskette in a specified device for a unique volume identifier and/or file label. If the volume identifier and/or file label are found on that diskette, a successful completion message is sent to the user who entered the command for that diskette. The diskette may be processed on the next diskette operation. If the correct diskette is not found, a message is sent to the user who entered the command. A check for this message in a CL program can direct the logic flow depending on whether the correct diskette is in the drive.

The diskette specified by the DEV parameter is searched. If a volume identifier is specified in the command, the volume identifier of the diskette is compared with the volume identifier in the command. If the correct volume identifier is found or if no volume identifier was specified in the command and a data file identifier is specified, the volume table of contents (VTOC) of the diskette is checked for the specified label.

If the correct data file identifier is found and the creation date is specified on the command, the date in the data file identifier is compared with that of the command. If they match, the correct file is found. If they do not match, the remaining labels in the VTOC are checked for both that data file identifier and the specified creation date.

If a match of the specified parameters is not found on the diskette, a message is sent to the user who entered the command.

Note: The identifiers are checked on each diskette in the following order:

1. Volume identifier
2. Data file identifier
3. The date it was created

Each parameter is checked on the diskette if (and only if) the parameters before it in the list are found on that diskette or were not specified in the command. If a match of all parameters in the command is not found on the diskette, a message is sent to the user who entered the command.

Since this command can be used in a CL program to determine whether the diskette is to be processed, the message for a media error while reading the VTOC is not sent to the user. Instead, a status message is sent to the user who entered the command. The status message can be checked by giving control to the CL program. If the status message is not checked, a message is sent to the user who entered the command.

Restriction: If a diskette has an extended label area, that extension area is not checked when searching for a file label. An informational message is sent notifying the user of this omission.

Note: Results when processing diskettes with labels that are not IBM standard labels are unpredictable. Initialize the diskette by specifying CHECK(*NO) on the Initialize Diskette (INZDKT) command.

Required Parameter

DEV

Specifies the name of the device in which the diskette being checked is located.

Optional Parameters

VOL

Specifies one or more volume identifiers used by the file.
| More information on this parameter is in Appendix A,
| "Expanded Parameter Descriptions."

***MOUNTED:** The volume currently placed in the device is used.

volume-identifier: Specify the volume identifier being compared with the volume identifier field on the diskette label. The identifier can have a maximum of 6 characters. Any combination of letters and numbers can be used. If the volume identifier does not match the diskette, an escape message is sent.

LABEL

Specifies whether a check is made for a specific file label on the diskette.

***NONE:** No data file identifier check is made.

data-file-identifier: Specify the data file identifier that is being checked for. A search for that data file identifier is done on the diskette in the specified device.

CRTDATE

Specifies whether the creation date of the data file identifier being checked is also checked.

***NONE:** The file creation date is not checked. If LABEL(*NONE) is specified, CRTDATE(*NONE) must also be specified.

creation-date: Specify the date that must match the creation date of the file being checked. The date must be entered in the format specified for the system values QDATFMT and, if separators are used, QDATSEP. When the correct file label is found, the creation date in that data file identifier is compared with the value in this parameter. If it does not match, the next data file identifier in the VTOC is checked.

Examples

Example 1: Checking a Volume Identifier

```
CHKDKT  DEV(QDKT) VOL(MASTER)
```

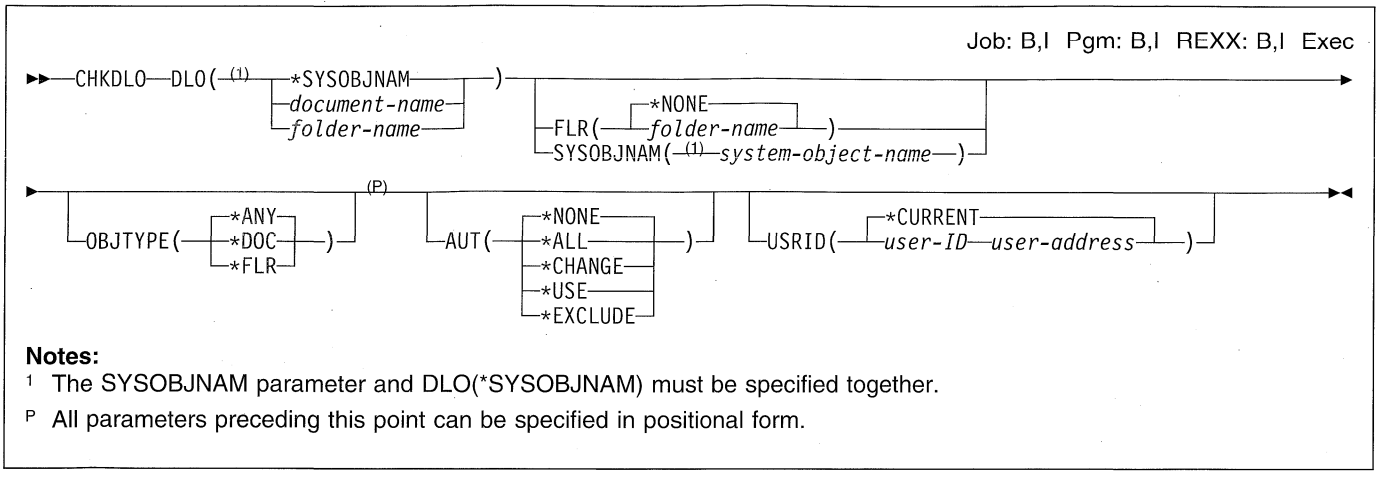
This command checks the volume identifier of the diskette in device QDKT for a volume identifier of MASTER. If the volume identifier of the diskette is MASTER, a completion message is sent. If the volume identifier of the diskette is not MASTER, a message is sent indicating that the volume identifier is incorrect and the job must be sent again.

Example 2: Checking Volume Identifier and File Creation Date

```
CHKDKT  DEV(QDKT)  VOL(VOLID)  LABEL(FILE)
  CRTDATE('7/4/76')
```

This command searches the diskette in device QDKT for a volume identifier of VOLID. If a diskette is found with that volume identifier, the file labels on the diskette are checked for a data file identifier of FILE. If that data file identifier is found and the creation date of the file is 7/4/76, the correct file and diskette have been found, and a completion message is sent. If the correct volume identifier and data file identifier and creation date are not found, a message is sent to the user indicating that the volume identifier is incorrect and the job must be sent again.

CHKDLO (Check Document Library Object) Command



Notes:
1 The SYSOBJNAM parameter and DLO(*SYSOBJNAM) must be specified together.
P All parameters preceding this point can be specified in positional form.

Purpose

The Check Document Library Object (CHKDLO) command verifies that an object exists and that a user has authority to the object before trying to access it.

These checks are useful before the user tries to access several objects at the same time. The CHKDLO command is also used to check the validity of document library object names contained in CL variables and to verify object authorizations under program control.

When the command runs, the system searches for the specified object. If the object is found, the system verifies that the user is authorized to that object in the manner specified on the CHKDLO command. If the object is not found or the user does not have the authority specified on the CHKDLO command, an escape message is sent to the user.

When the CHKDLO command is used in a CL program, at least one Monitor Message (MONMSG) command should follow the CHKDLO command to monitor messages that result from running this command. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

Required Parameter

DLO

Specifies the document library object that is checked.

***SYSOBJNAM:** The system object name is used to identify the document or folder that is checked. This parameter must be used to check a document that is not in a folder. It may also be used instead of a folder name or document name, whenever the system object name is known. The SYSOBJNAM parameter and DLO(*SYSOBJNAM) must be specified together.

document-name: Specify the name of the document that is checked.

folder-name: Specify the name of the folder that is checked.

Optional Parameters

FLR

Specifies the name of the folder that contains the document.

***NONE:** The object that is checked is not contained in a folder.

folder-name: Specify the name of the folder that contains the document or folder that is checked. A folder name can be specified only if a folder or document name is specified for the DLO parameter.

SYSOBJNAM

Specifies the system object name. This parameter is valid only when DLO(*SYSOBJNAM) or DOCL(*SYSOBJNAM) is specified. A full ten characters must be specified.

OBJTYPE

Specifies the OS/400 system object type of the document library object that is checked. OBJTYPE(*DOC) cannot be specified when a document or folder name is specified for the DLO parameter, and FLR(*NONE) is also specified. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***ANY:** The object that is checked is either a document or a folder.

***DOC:** The object that is checked is a document.

***FLR:** The object that is checked is a folder.

AUT

Specifies the authority being checked. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***NONE:** Authority is not checked.

***ALL:** The user is checked for authority to perform all operations on the document or folder except those limited to the owner.

***CHANGE:** The user is checked for authority to perform all operations on the document or folder except those limited to the owner or 'ALL' authority.

***USE:** The user can perform basic operations on the document library object, such as running a program or reading a file. The user cannot change the document library object. *USE authority provides object operational authority and read authority.

***EXCLUDE:** The user cannot access the document library object.

USRID

Specifies the user ID and address of the user for whom the object is being checked. If a user ID and address of someone other than the user who is signed on is specified, both users must be enrolled in the system directory. The user who is signed on must be granted permission to work on behalf of the specified user.

***CURRENT:** The user profile under which the current job is running is used.

Element 1: User ID

user-ID: Specify the user ID of the user on whose behalf the object is to be checked.

Element 2: User Address

user-address: Specify the address of the user on whose behalf the object is to be checked.

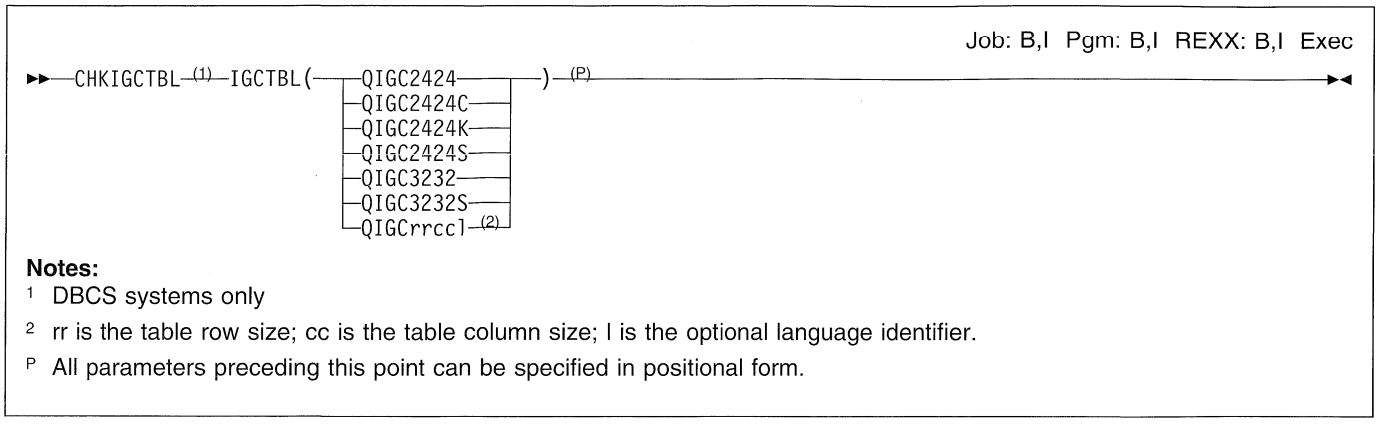
Note: The USRID parameter is useful only when the AUT parameter and any special value are specified, except *NONE.

Example

```
CHKDLO  DLO(FLR1)  OBJTYPE(*ANY)  AUT(*NONE)
  USERID(USER1 ADDR1)
```

This command checks for the existence of a folder named FLR1 on behalf of a user whose user ID is USER1 and whose address is ADDR1. The user's authority to FLR1 is not checked.

CHKIGCTBL (Check DBCS Font Table) Command



Notes:

1 DBCS systems only

2 rr is the table row size; cc is the table column size; l is the optional language identifier.

P All parameters preceding this point can be specified in positional form.

Purpose

The Check DBCS Font Table (CHKIGCTBL) command checks for the existence of a specified double-byte character set (DBCS) font table. Use this command to verify that one of the DBCS font tables in the system prints and shows characters in the matrix pattern used by a given device. If the table does not exist, the system sends you a message. If the table exists, the system does not send you a message.

DBCS font tables contain the images in a given dot matrix of the DBCS extension characters used on the system. The system refers to the tables when printing and showing these characters. There are separate tables for each character image matrix used by devices attached to the system.

Required Parameter

IGCTBL

Specifies the name of the DBCS font table for which its existence is being checked. Choose one of the following table names:

QIGC2424: The Japanese DBCS font table used for showing and printing extension characters in a 24-by-24 dot matrix image.

QIGC2424C: The Traditional Chinese DBCS font table used for printing extension characters in a 24-by-24 dot matrix image.

QIGC2424K: The Korean DBCS font table used for printing extension characters in a 24-by-24 dot matrix image.

QIGC2424S: The Simplified Chinese DBCS font table used for printing extension characters in a 24-by-24 dot matrix image.

QIGC3232: The Japanese DBCS font table used for showing and printing extension characters in a 32-by-32 dot matrix image.

QIGC3232S: The Simplified Chinese DBCS font table used for printing extension characters in a 32-by-32 dot matrix image.

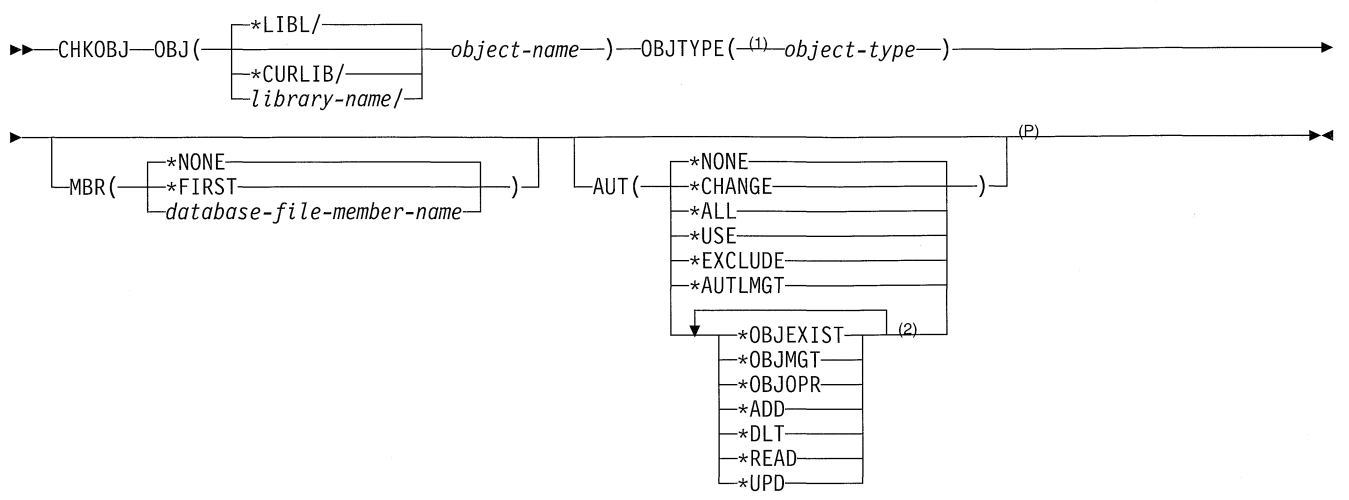
QIGCrrccl: The name of the DBCS font table checked for must always be in the format QIGCrrccl, where *rr* is the table row matrix size, *cc* is the table column matrix size, and *l* is an optional language identifier.

Example

```
CHKIGCTBL   IGCTBL(QIGC2424)
```

This command causes the system to check for the Japanese DBCS font table that contains character images in a 24-by-24 dot matrix image.

CHKOBJ (Check Object) Command



Notes:

1 A list of the valid OS/400 object types for this command is in Appendix A.

2 A maximum of 7 repetitions, each to be used only once.

| P All parameters preceding this point can be specified in positional form.

Purpose

The Check Object (CHKOBJ) command verifies that an object exists and that a user has authority to the object before access to it is permitted. For verification, as many as seven specific authorities can be specified in the command.

These checks are particularly useful before the user tries to access several objects at the same time. The CHKOBJ command is also used to check the validity of object names contained in CL variables and to verify object authorizations under program control.

When the command runs, the system searches for the specified object. If the object is found, the system verifies that the user is authorized to that object as specified on the CHKOBJ command. If the object is not found or the user does not have the authorities specified on the CHKOBJ command, an escape message is sent to the user.

When the CHKOBJ command is used in a CL program, at least one Monitor Message (MONMSG) command should follow the CHKOBJ command to monitor for any messages that result from running this command. For the list of error messages that can be monitored for each command, refer to *Programming Reference Summary*.

Required Parameters

OBJ

Specifies the qualified name of the object being checked. If no library name is given, *LIBL is used to find the object.

| The name of the object can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

object-name: Specify the name of the object that is checked.

OBJTYPE

Specifies the object type of the OS/400 system object being checked. Enter the predefined value that specifies the object type. More information on this parameter is in | Appendix A, "Expanded Parameter Descriptions."

Optional Parameters

MBR

Specifies, if a member of a database file is being checked, the name of the file member.

Note: For a specified logical file member, the data authorities specified by AUT are checked for

each of the physical file members on which the logical file member is based.

***NONE:** Database file members are not checked, but the existence and (optionally) the authority of the file are checked. For all other object types (including device files), *NONE is the only valid value for the MBR parameter. If *NONE is specified, and if data authorities are tested for a logical file, data authorities are ignored when the user's authority to the file is being checked.

***FIRST:** The first member in the database file is used.

database-file-member-name: Specify the name of a physical or logical file member to be checked by the CHKOBJ command. Values specified for the OBJ and OBJTYPE parameters must identify a database file and the member specified must be a member of the database file specified in the OBJ parameter.

AUT

Specifies the authority being checked.

***NONE:** The user's authority is not checked.

***CHANGE:** The user's object operational authority and all data authorities to the object are checked regardless of the object type.

***ALL:** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence and specify the security for the object, change the object, and perform basic functions on the object. The user can change ownership of the object.

***USE:** The user can perform basic operations on the object, such as running a program or reading a file. The user cannot change the object. *USE authority provides object operational authority and read authority.

***EXCLUDE:** The user cannot access the object.

***AUTLMGT:** The user's authority to add, delete, or change users and their authorities on the authorization list or delete the authorization list is checked.

Note: The OBJTYPE(*AUTL) parameter must be specified before specifying AUT(*AUTLMGT).

***OBJEXIST:** The user's authority to control object ownership and existence is checked. These authorities are required for a user to delete an object; to free storage; or to save, restore, or transfer ownership of the object. A user with special save system (*SAVSYS) authority does not need object existence authority to save or restore the object.

***OBJMGT:** The user's authority to manage the access to, and availability of, the object is checked. A user with object management authority can check personal authority status, move and rename objects, and add members to database files.

***OBJOPR:** The user's authority to manage access to, and availability of, the object is checked. Object operational authority has no data authority associated with it. Data authorities (listed below) are individually checked.

***ADD:** The user's add authority, which is needed to add entries to the object (for example, adding job entries to a queue or adding records to a file) is checked.

***DLT:** Delete authority allows the user to remove entries from an object, for example, remove messages from a message queue or records from a file.

***READ:** The user's read authority, which is needed to retrieve the contents of the object entry is checked.

***UPD:** The user's update authority, which is needed to update entries in the object is checked.

Examples

Example 1: Checking for Existence of a Program

```
CHKOBJ  OBJ(LIB1/PROG1)  OBJTYPE(*PGM)
```

This command checks for the existence of a program named PROG1 in library LIB1. The user's authorities to PROG1 are not checked.

Example 2: Checking for User's Authority to File

```
CHKOBJ  OBJ(SOURCE1)  OBJTYPE(*FILE)
  MBR(MBR3)  AUT(*CHANGE)
```

This command checks for the existence of file SOURCE1 and for the existence of member MBR3 in file SOURCE1. It also checks to see if the user has *CHANGE authority to file SOURCE1.

Example 3: Checking for User's Authority to Program

```
CHKOBJ  OBJ(LIB1/PROG1)  OBJTYPE(*PGM)  AUT(*CHANGE)
```

This command checks the existence of program PROG1 in library LIB1. It also checks to see if the user has *CHANGE authority to PROG1.

Messages that can be monitored by the Monitor Message (MONMSG) command if sent by the CHKOBJ command are:

CPF9801	OBJECT NOT FOUND–PROG1 does not exist.
CPF9802	OBJECT NOT AUTHORIZED–The user that issued this command does not have *CHANGE authority to PROG1.
CPF9810	LIBRARY NOT FOUND–LIB1 cannot be located.
CPF9820	NOT AUTHORIZED TO LIBRARY–The user that issued this command is not authorized to the library named LIB1.
CPF9830	UNABLE TO ALLOCATE LIBRARY–The library named LIB1 is locked and cannot be accessed.

Example 4: Checking User's Authority to a Logical File Member

```
CHKOBJ  OBJ(FILEA)  OBJTYPE(*FILE)
  MBR(MBR1)  AUT(*USE)
```

This command checks the user's authority to use logical file member MBR1, and each physical file member on which MBR1 is based.

In addition to the messages listed in the previous example, messages that can be monitored by the MONMSG command if sent by the CHKOBJ command, are:

CPF9815 MEMBER IN FILE NOT FOUND–MBR1 cannot be found in FILEA or FILEA does not contain members. If FILEA is a device file, a CPF9899 message is sent.

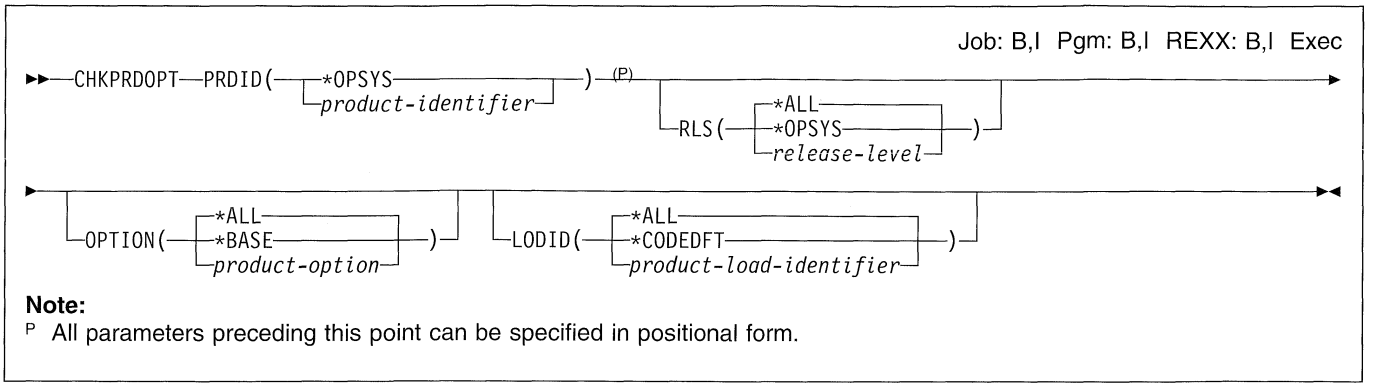
CPF9899 FUNCTION NOT PERFORMED–This message is a summary escape message that is always preceded by a diagnostic message. If FILEA is a device file, message CPF2168 precedes message CPF9899. If FILEA is locked, message CPF3202 precedes this message.

Example 5: Checking User's Add and Delete Authority

```
CHKOBJ   OBJ(FILEA)  OBJTYPE(*FILE)  MBR(MBR1)
  AUT(*ADD  *DLT)
  MONMSG   MSGID(CPF9802)  EXEC(GOTO  ERROR1)
```

These two commands (CHKOBJ and MONMSG) are used to verify that the user has both add and delete authority for each of the physical file members on which the logical file member MBR1 in the logical file FILEA is based. If the user does not have both of the data authorities for all of the based-on physical file members, escape message CPF9802 is sent to the program, and control in the program is passed to the command that has the label ERROR1.

CHKPRDOPT (Check Product Option) Command



Note:
P All parameters preceding this point can be specified in positional form.

Purpose

The Check Product Option (CHKPRDOPT) command reports differences between the correct structure and the actual structure of a software product. (For example, if an object is deleted from an installed product, running this command will report the error.) Use the informational and diagnostic messages to determine the condition of the product.

Note: Running this command does not necessarily issue an escape message if the product has been deleted or is being created.

Required Parameter

PRDID
Specifies the identifier of the software product being checked.

***OPSYS:** The OS/400 licensed program is checked.

product-identifier: Specify a product identifier. The identifier must be seven characters in length.

Optional Parameters

RLS
Specifies the release level of the product to be checked.

***ALL:** All releases of the product are checked.

***OPSYS:** The release level of the product being checked is the same as the release level of the operating system currently installed.

release-level: Specify the release level in the format VxRxMx, where Vx is the version number, Rx is the release number, and Mx is the modification number.

OPTION
Specifies the product option being checked.

***ALL:** All options of the product are checked.

***BASE:** The base option of the product is checked.

product-option: Specify an option number ranging from 1 through 99.

LODID
Specifies the product load being checked.

***ALL:** All product loads for a given option are checked.

***CODEDFT:** The code load is checked.

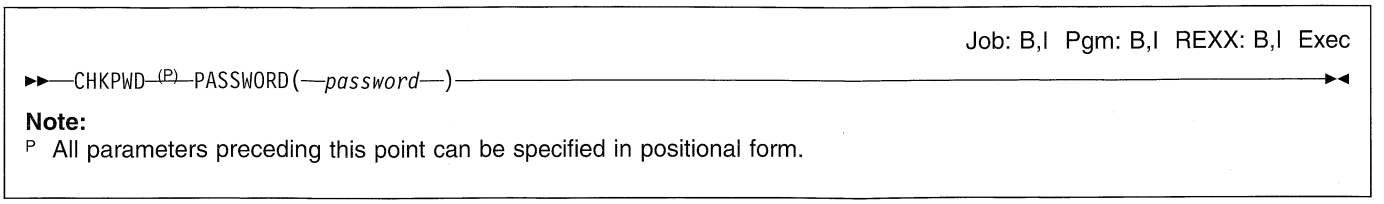
product-load-identifier: Specify the product load identifier. The load identifier must be four characters in length.

Example

```
CHKPRDOPT   PRDID(5738AP1)
```

This command checks all releases of product 5738AP1.

CHKPWD (Check Password) Command



Purpose

The Check Password (CHKPWD) command checks a password and determines its validity. The password is the security key that allows the user to sign on the system.

Required Parameter

PASSWORD

Specifies a password value that is checked for validity. If the password is correct, no message is shown. If the password not correct, one of the following messages is shown after each attempt:

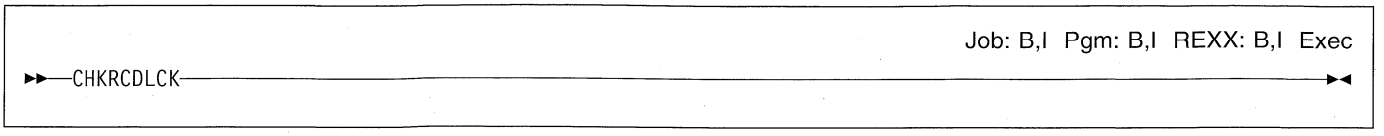
CPF2362	Password not correct.
CPF2363	One attempt left to check your password.
CPF2364	The number of allowable attempts to check your password has been exceeded.

Example

```
CHKPWD  JOHNJONES
```

This command checks whether the current password is JOHNJONES.

CHKRCDLCK (Check Record Locks) Command



Purpose

The Check Record Locks (CHKRCDLCK) command detects whether a job has any record locks. This command is used to detect whether the job has any record locks before transferring to another group job while the user is in the middle of changing the database. This command can be used exclusively to check record locks on a job, and then the user can exit the command.

The CHKRCDLK command sends an escape message if there are any record locks currently held by the job that issued the command.

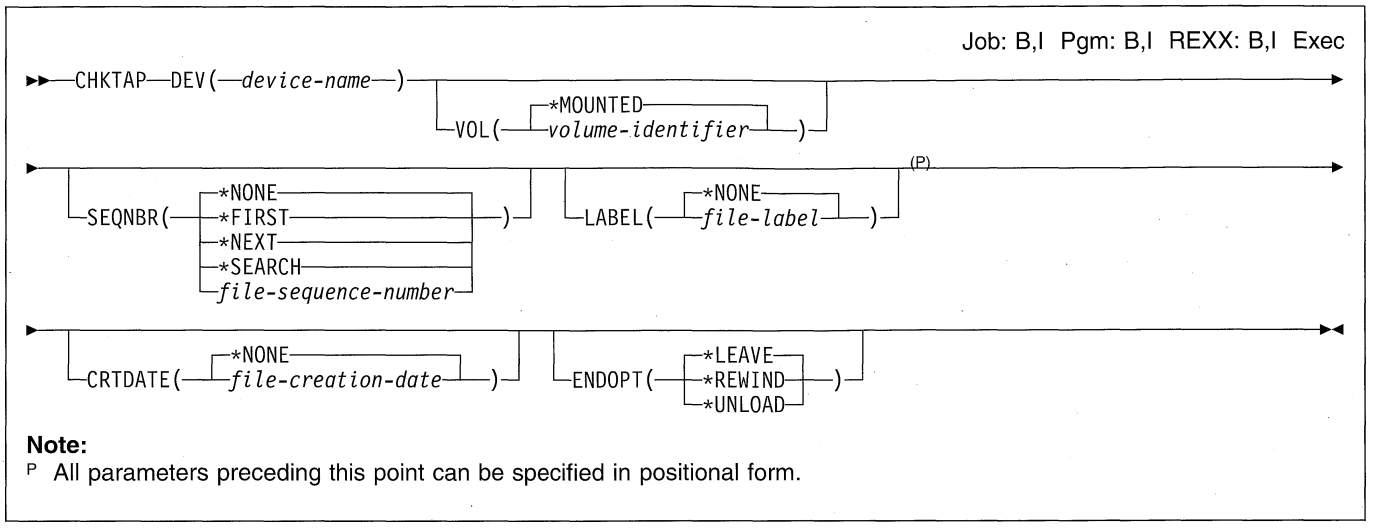
There are no parameters for this command.

Example

```
CHKRCDLCK
```

This command sends an escape message if there are any record locks held by the job.

CHKTAP (Check Tape) Command



Purpose

The Check Tape (CHKTAP) command searches a volume on the specified device for a unique volume identifier and/or file label. If the correct tape is on the device, the user may process this file on the next tape operation by specifying the same sequence number that was specified in the CHKTAP command. If the correct tape is not found, an escape message is sent, and the tape is rewound regardless of the end option specified. A check for this message in a CL program can direct the logic flow if the correct tape is on the device.

If a volume identifier is specified in the command, the volume identifier of the tape is compared with the volume identifier of the command. If a match is not found on the tape, an escape message is sent. This message contains the volume identifier found on the tape. If the correct volume identifier is found or if no volume identifier is specified in the command, and a sequence number is specified, that sequence number is located on the tape. For a standard labeled tape, the sequence number is determined from the header label of the file. For a tape that is not labeled, the sequence number is determined from the number of tape markers from the beginning-of-tape marker. If the sequence number is not found, an escape message is sent.

If the sequence number specified on the command is found and a label was specified, the file identifier in the header label is compared with the value in the LABEL parameter. The LABEL parameter is valid only for a standard-label tape. If the tape is a standard-label tape and the file label at that sequence number does not match the value in the LABEL parameter, an escape message is sent. This message contains the date the file was created and the file label for the file at that sequence number. If a match is found and a date is not specified in the command, the file is found. If the correct file identifier is found and a date is specified in the command, the date in the header label is compared with that of the command. If a match exists, the correct file is found. If there is not a match, an escape message is sent. This message contains the file label and the date that the file at that sequence number was created.

If SEQNBR (*SEARCH) is specified, a value for the LABEL parameter must be specified. The file label for each file on tape is checked for the LABEL parameter until a match is found. If the file is not found, an escape message is sent.

If the sequence number specified on the command is found and a LABEL parameter was not specified but the CRTDATE parameter was, the date in the header label for the file at the sequence number is compared with the date value in the CRTDATE parameter. If the dates do not match, an escape message is sent. The CRTDATE is valid only for standard label tape.

Note: The values in the command parameters are compared with the values on the tape in the following order:

1. Volume identifier
2. Sequence number
3. File identifier in the header label at the sequence number specified
4. File creation date in the header label at the sequence number specified

The tape is checked for each parameter only if the parameters before it in the list are found on the tape or were not specified in the command.

Required Parameter

DEV

Specifies the name of the device where the volume is being checked.

Optional Parameters

VOL

Specifies one or more volume identifiers used by the file. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***MOUNTED:** The volume currently placed in the device is used.

volume-identifier: Specify the volume identifier of the labeled volume. The volume identifier read from the tape is compared to this value. If the volume identifier specified is not found on the tape, an escape message is sent.

SEQNBR

Specifies whether a check is made for a specific sequence number of a data file on the tape. For standard-labeled tapes, this is the sequence number in the file header label. For tapes that are not labeled, this is the sequence number determined from the number of tape markers from the beginning of tape.

***NONE:** No check is made for a file on this volume.

***FIRST:** A check is made for the first file on this volume.

***NEXT:** A check is made for the next file on this volume. At the current sequence number is at the beginning of the volume, this value checks the first file on that volume.

***SEARCH:** A data file is searched for that has an identifier that matches the LABEL parameter value. If the last tape operation on the device specified ENDOPT(*LEAVE), the file search begins with the first data file beyond the current tape position. If ENDOPT(*LEAVE) was not used for the last tape operation, or if the tape was manually rewound since an ENDOPT(*LEAVE) operation, the search begins with the first data file on the volume. SEQNBR(*SEARCH) cannot be specified when LABEL(*NONE) is also specified, and cannot be used to check a volume that is not labeled. An escape message is sent if the file is not found.
If the sequence number is not found on the tape, an escape message
is sent.

file-sequence-number: Specify the sequence number of the file that is used.

LABEL

Specifies whether a label identifier is checked. If a label is specified, a sequence number must be specified for the SEQNBR parameter.

***NONE:** No check is made for a label identifier on the tape. LABEL(*NONE) and CRTDATE(*NONE) must be specified for a volume that is not labeled.

file-label: Specify the identifier (17 characters maximum) of the data or save/restore file to check. If a label is specified, a sequence number must be specified for the SEQNBR parameter. The file identifier of the file at that sequence number is compared with the label identifier specified by this parameter. If the label does not match, a message is sent.

CRTDATE

Specifies whether the date on which the file was created is checked. If SEQNBR(*NONE) is specified, CRTDATE(*NONE) must also be specified.

***NONE:** The date on which the file was created is not checked. CRTDATE(*NONE) and LABEL(*NONE) must be specified for a volume that is not labeled.

file-creation-date: Specify the date that must match the date of the file being checked. The date must be entered in the format specified for the system values QDATFMT and, if separators are used, QDATSEP. If the date of the file being checked does not match the date specified by this parameter, an escape message is sent.

ENDOPT

Specifies the operation that is automatically performed on the tape volume after the operation ends. If more than one volume is included, this parameter applies only to the last tape volume used; all other tape volumes are rewound and unloaded when the end of the tape is reached.

***LEAVE:** The tape does not rewind or unload after the operation ends. It remains at the current position on the tape drive.

***REWIND:** The tape is rewound, but not unloaded.

***UNLOAD:** The tape is rewound and unloaded.

Examples

Example 1: Checking the Volume Identifier

```
CHKTAP  DEV(TAPE1)  VOL(TAPEVOL)
```

This command checks the volume identifier of the volume on the tape device TAPE1. If the volume identifier on the tape is TAPEVOL, the command completes normally and no message is sent. If the volume identifier on the tape is not TAPEVOL, an escape message is sent.

Example 2: Checking for a Specific Sequence Number

```
CHKTAP  DEV(TAPE2)  VOL(VOLID)  SEQNBR(5)
  LABEL(FILE5)  CRTDATE('1/9/84')
```

This command checks the volume on the tape device TAPE2 for a volume identifier of VOLID. If that volume is found, sequence number 5 is located on the tape (it must be a standard-labeled tape). The sequence number in the file label is used to position to sequence number 5. If the sequence number is found and the header label contains both the file identifier FILE5 and the date of 1/9/84, the correct tape and file has been found, and a completion message is sent. The next tape operation can specify sequence number 5 to access this file without positioning the

tape. If the specified volume is not found or the tape is not a standard labeled volume, an escape message is sent. If the volume is found and the sequence number is not found, an escape message is sent. If the file label at that sequence number is not FILE5, an escape message is sent. If the date at that sequence number is not 1/9/84, an escape message is sent.

CLOF (Close File) Command

Job: B,I Pgm: B,I REXX: B,I Exec

Note:
(P) All parameters preceding this point can be specified in positional form.

Purpose

The Close File (CLOF) command closes a database file opened with the Open Database File (OPNDBF) or Open Query File (OPNQRYF) commands.

Restriction: This command is only used to close a file that is opened by the OPNDBF or OPNQRYF command.

Required Parameter

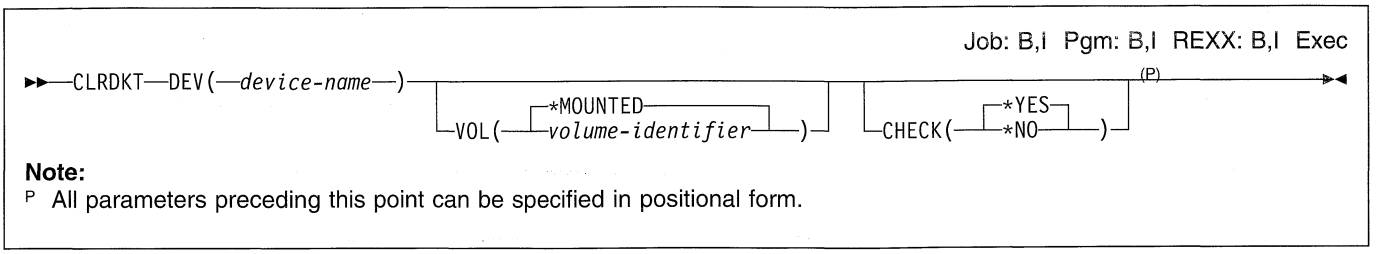
OPNID
Specifies the name used on the Open Query File (OPNQRYF) or the Open Database File (OPNDBF) for identifying this open identifier (OPNID). This OPNID is specified when closing this file. It cannot be reused without first closing this file.

Example

```
CLOF  OPNID(APPL)
```

This command closes a database file that was opened with APPL as the OPNID. The file was previously opened using the OPNDBF or OPNQRYF command with APPL specified (or defaulted) as the OPNID.

CLRDKT (Clear Diskette) Command



Purpose

The Clear Diskette (CLRDKT) command deletes all active and inactive files from a diskette by deleting the data file identifiers from the diskette label area. A single (expired) file is defined, covering the entire diskette, and is identified as DATA. The data contained in the files is not deleted. Refer to the DLTDKTLBL (Delete Diskette Label) command and the INZDKT (Initialize Diskette) command to delete the data in the files.

The CLRDKT command does not test the diskette for defects nor does it change the volume identifier and owner identifier fields. The error map also does not change.

The diskette in the specified device (DEV parameter) is cleared by the CLRDKT command. If no volume identifier is specified, the command clears the diskette in the specified device. If an identifier is specified and it is the same as the identifier on the diskette in the specified device, the diskette is cleared.

Note: When processing diskettes with labels that are not IBM standard labels, you may get unpredictable results. To initialize the diskette, enter the Initialize Diskette (INZDKT) command, with CHECK(*NO) specified.

Restriction: A diskette that has an extended label area cannot be cleared; it must be initialized by the Initialize Diskette (INZDKT) command.

Required Parameter

DEV

Specifies the name of the device in which the diskette being cleared is located.

Optional Parameters

VOL

Specifies one or more volume identifiers used by the file. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***MOUNTED:** The volume currently placed in the device is used.

volume-identifier: Specify a volume identifier to be compared with the diskette label volume identifier field of the diskette being cleared. The identifier can have up to 6 characters. Any combination of letters and numbers can be used.

CHECK

Specifies whether to check the diskette in the specified device for active files before it is cleared. Active files are files having an expiration date later than the system date.

***YES:** A check is performed on files whose labels are in cylinder 0 only. File labels in an extended file label area (not supported by the AS/400 system) are not checked. If any active files are found on a diskette, a message is sent to the system operator. The operator can continue the clear function, and erase the active files, or he can end the operation.

***NO:** The diskette is cleared without being checked for active files.

Examples

Example 1: Clearing Diskette with Volume Identifier of MASTER

```
CLRDKT   DEV(QDKT)  VOL(MASTER)  CHECK(*NO)
```

This command clears the diskette in device QDKT if its volume identifier is MASTER.

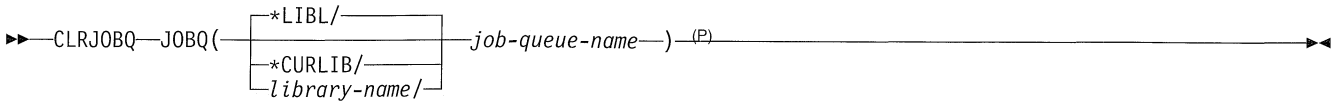
Example 2: Checking for Active Files Before Diskette is Cleared

```
CLRDKT   DEV(DKT1)
```

This command clears the diskette in device DKT1. Because VOL(*MOUNTED) is assumed, the diskette could be in either the basic exchange or save/restore formats, and a volume identification check is not made. However, because CHECK(*YES) is also assumed, the diskette is checked for active files before it is cleared.

CLRJOBQ (Clear Job Queue) Command

Job: B,I Pgm: B,I REXX: B,I Exec



Note:

(P) All parameters preceding this point can be specified in positional form.

Purpose

The Clear Job Queue (CLRJOBQ) command removes all the batch jobs (including jobs that have been held) from the specified job queue. Any jobs that are currently being read in and interactive jobs that have been rerouted to the job queue remain on the queue. The running of jobs that were started from the job queue is not affected.

Required Parameter

JOBQ

Specifies the qualified name of the job queue that is cleared of all waiting or held jobs.

The name of the job queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

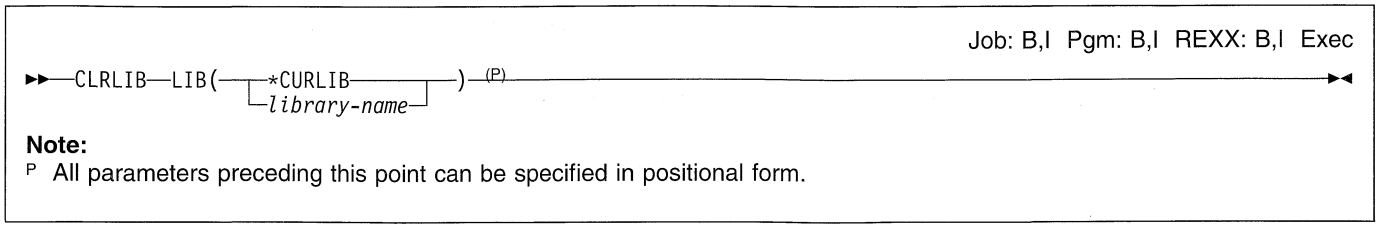
job-queue-name: Specify the name of the job queue.

Example

```
CLRJOBQ   JOBQ(QBATCH)
```

This command removes all jobs currently in the IBM-supplied job queue, QBATCH. Any job currently being read in is not affected.

CLRLIB (Clear Library) Command



Note:
P All parameters preceding this point can be specified in positional form.

Purpose

The Clear Library (CLRLIB) command deletes all of the objects from the specified library that a user has the authority to delete. The CLRLIB command does not delete the specified library, only the objects for which the user has object existence authority; the other objects remain in the library. Objects being used by any other job when this command is entered are not deleted.

Restrictions:

1. The user must have object existence authority for every object being deleted.
2. This command cannot be used to clear the QSYS, QSPL, or QRECOVERY libraries.

Required Parameter

LIB

Specifies the name of the library that is cleared of all objects for which the user has object existence authority. Objects for which the user does not have object existence authority remain in the library.

***CURLIB:** The current library for the job is cleared. If no library is specified as the current library for the job, the QGPL library is used.

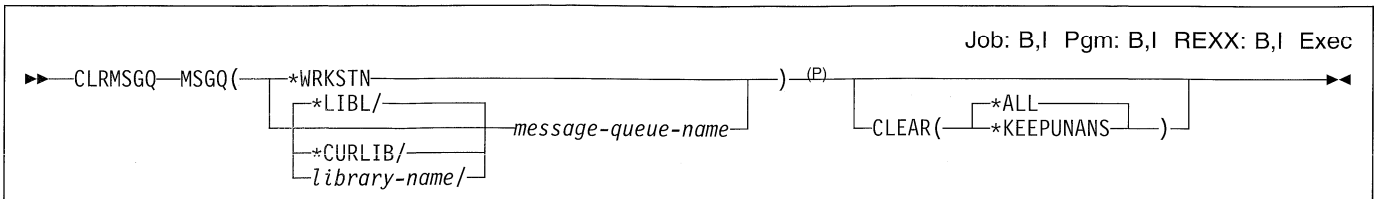
library-name: Specify the name of the library to be cleared.

Example

```
CLRLIB   LIB(A)
```

This command deletes all objects in library A that are not in use and for which the user has object existence authority.

CLRMSGQ (Clear Message Queue) Command



Note:

(P) All parameters preceding this point can be specified in positional form.

Purpose

The Clear Message Queue (CLRMSGQ) command clears (removes) all messages from a specified message queue. Once cleared, the data cannot be shown or printed. If the specified message queue is not allocated to a job, it is implicitly allocated by this command for the duration of the command. If the specified message queue is *WRKSTN or a work station message queue, it is not allocated and the message queue is cleared even if the work station device description is allocated to another job.

Required Parameter

MSGQ

Specifies the qualified name of the message queue that is cleared. If a specific message queue name is specified (instead of a generic name), only the first message queue found with that name is cleared.

***WRKSTN:** The work station message queue is cleared. This is not allowed in batch mode.

The name of the message queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

message-queue-name: Specify the name of the message queue that is cleared.

Optional Parameter

CLEAR

Specifies which messages are cleared from the message queue.

***ALL:** All messages are cleared from the message queue. If there are unanswered messages on the queue, the default reply is returned to the sender before the message is cleared.

***KEEPUNANS:** Unanswered inquiry messages remain on the specified message queue. Other messages are removed.

Examples

Example 1: Clearing All Messages

```
CLRMSGQ   MSGQ(*CURLIB/MQFIN)
```

This command clears all messages from a message queue named MQFIN, which is located in the current library for the job.

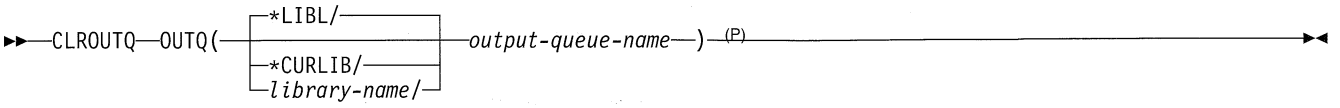
Example 2: Keeping Unanswered Messages

```
CLRMSGQ   MSGQ(*CURLIB/MQFIN)  CLEAR(*KEEPUNANS)
```

This command clears all messages except unanswered inquiry messages from a message queue called MQFIN, which is located in the current library for the job.

CLROUTQ (Clear Output Queue) Command

Job: B,I Pgm: B,I REXX: B,I Exec



Note:

(P) All parameters preceding this point can be specified in positional form.

Purpose

The Clear Output Queue (CLROUTQ) removes spooled files from the specified output queue. The CLROUTQ command removes all spooled files that are waiting to be written on an output device from the specified queue, including files that are in the hold state. Spooled files that are currently being produced by programs or that are being written to an output device are not removed from the queue.

Required Parameter

OUTQ

Specifies the qualified name of the output queue.

The name of the output queue can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

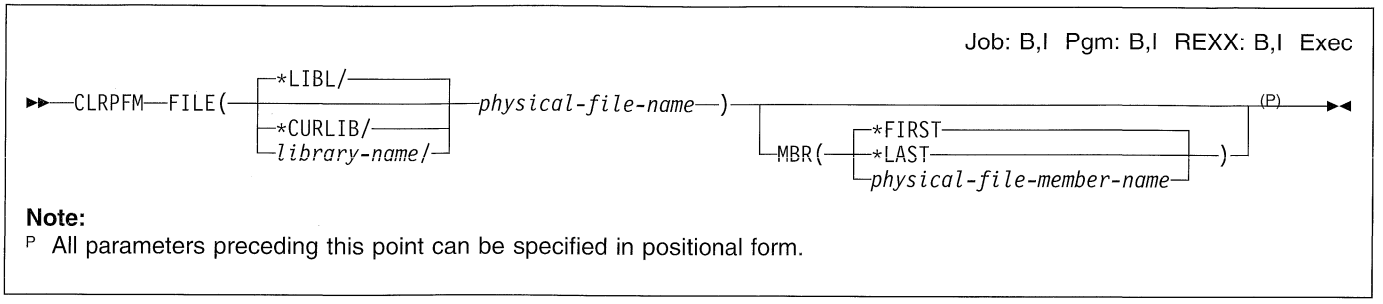
output-queue-name: Specify the name of the output queue that is cleared.

Example

```
CLROUTQ   OUTQ(QPRINT)
```

This command removes the entries for all spooled files from the output queue, QPRINT, that are waiting to be printed or are being held. The entries for the file currently being printed and files still receiving data from programs that are currently running are not affected.

CLRPFM (Clear Physical File Member) Command



Purpose

The Clear Physical File Member (CLRPFM) command removes all the data (including deleted records) from the specified member of a physical file. If ALLOCATE(*NO) is specified when the file is created, the record count for the member is set to zero, and the member size is set to the minimum size possible. If ALLOCATE(*YES) is specified when the file is created, the CLRPFM command resets the member size to the value used when the file is initially created. For more information, refer to the ALLOCATE parameter for the Create Physical File (CRTPF) command. An attempt to get a record from the cleared member results in an error message being sent to the user or program that made the attempt.

Note: The CLRPFM command ignores all file overrides that are currently in effect for the job.

Restriction: The user must have object operational, object management, and delete authority for the physical file that contains the member. If any of the access paths to the member are in use when this command is entered, or if the physical file member is in use, the command is not run. An *EXCL lock is required on the member to clear it.

Required Parameter

FILE

Specifies the qualified name of the physical file that contains the member being cleared.

The name of the physical file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

physical-file-name: Specify the name of the physical file.

Optional Parameter

MBR

Specifies the name of the member, or the first member, that is cleared.

***FIRST:** The first member in the database file is used.

***LAST:** The last member of the specified physical file is cleared.

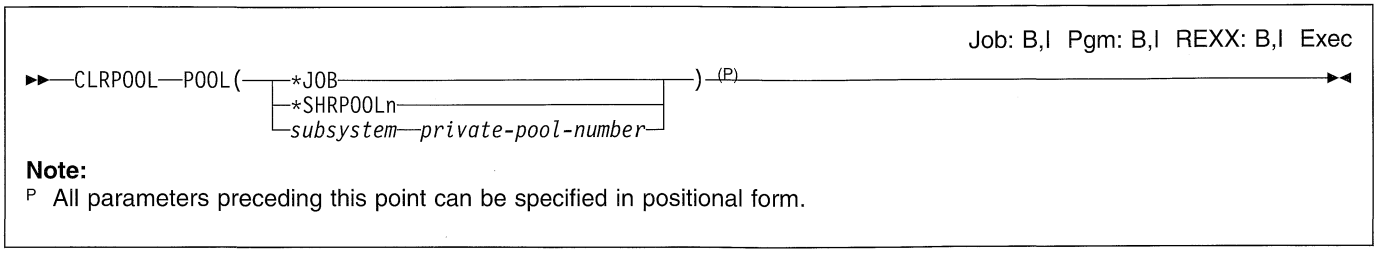
physical-file-member-name: Specify the name of the physical file member that is cleared.

Example

```
CLRPFM   FILE(*CURLIB/INV)  MBR(FEB)
```

This command clears the member named FEB in the physical file INV, found in the current library for the job *CURLIB. It is not cleared until all jobs currently using the member and all jobs using the access paths over the member are done with it.

CLRPOOL (Clear Pool) Command



Note:

| P All parameters preceding this point can be specified in positional form.

Purpose

The Clear Pool (CLRPOOL) command clears all objects from a main storage pool. This allows the Set Object Access (SETOBJACC) command to report on storage usage within a pool.

Required Parameter

POOL

Specifies the pool to be cleared of all objects.

***JOB:** The pool associated with the job is cleared.

***SHRPOOLn:** A general-purpose shared pool is cleared. Valid values range from 1 through 10.

Element 1: Subsystem

subsystem: Specify a subsystem name.

Element 2: Pool Identifier

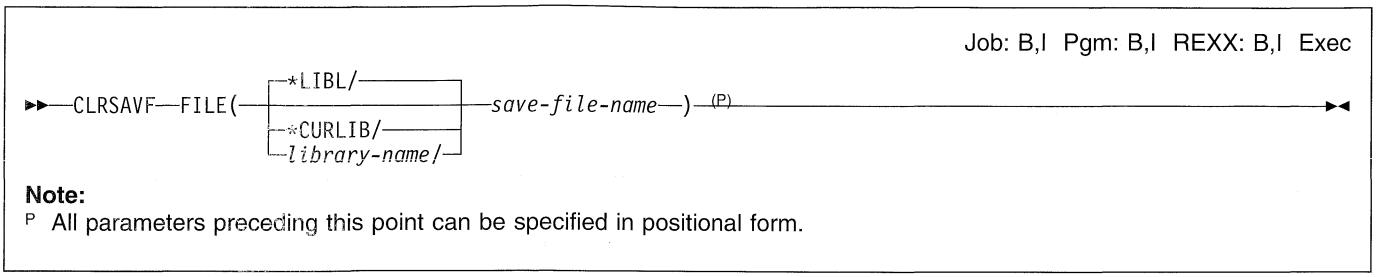
pool-identifier: Specify a subsystem pool identifier.

Example

```
CLRPOOL    POOL(*JOB)
```

This command clears the pool associated with the job in which the command was processed.

CLRSAVF (Clear Save File) Command



Note:
| (P) All parameters preceding this point can be specified in positional form.

Purpose

The Clear Save File (CLRSAVF) command clears the contents of a save file. This command clears all existing records from the save file and reduces the amount of storage used by this file.

A save file must be cleared before it can be used again to receive data from a save command or to receive another save file. If the user attempts to write new save data into a save file that already contains records, an inquiry message is sent to the work station for an interactive job, or to the system operator for a batch job, unless a save command is used and CLEAR(*ALL) is specified.

Note: This command ignores all file overrides that are currently in effect for the job.

Restriction: The user of this command must have operational and object management authority for the save file and read authority for the specified library.

Required Parameter

FILE

Specifies the qualified name of the save file to clear.

| The name of the save file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

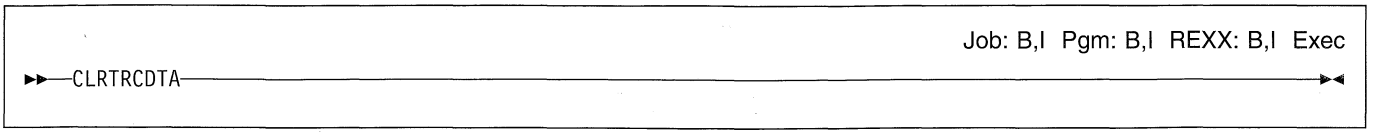
save-file-name: Specify the name of the save file to clear.

Example

```
CLRSAVF   FILE(ONLINE)
```

This command clears the contents of save file ONLINE. Any existing records in the file are removed, and the file size is reduced to the minimum size possible.

CLRTRCDTA (Clear Trace Data) Command



Purpose

The Clear Trace Data (CLRTRCDTA) command clears (removes) all of the data from any previous trace operations in this debugging session. Once cleared, the data can no longer be displayed or printed.

Restriction: This command is valid only in debug mode. To start debug mode, refer to the STRDBG (Start Debug) command.

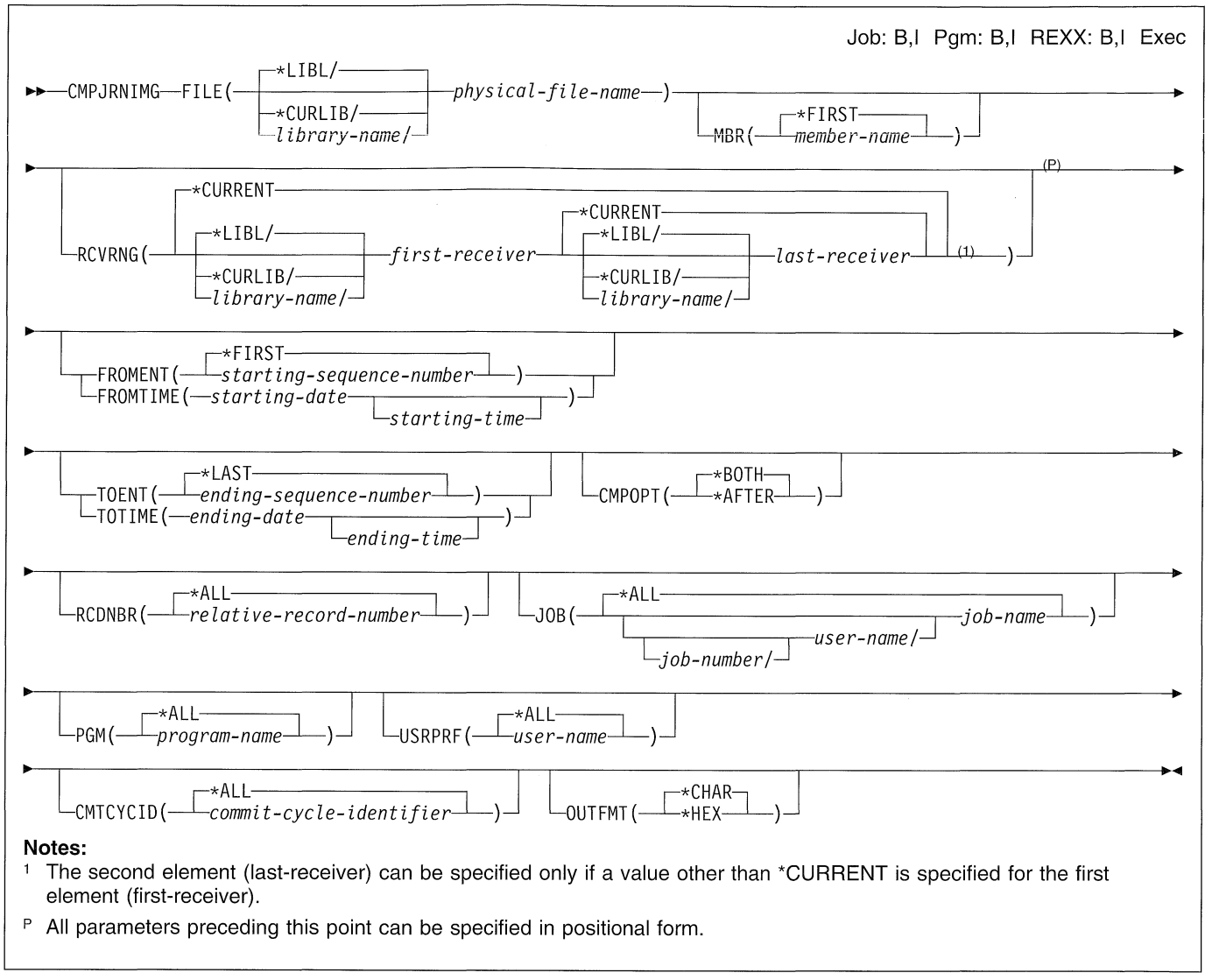
There are no parameters for this command.

Example

```
CLRTRCDTA
```

This command clears all of the data recorded from any and all previous tracing operations in all of the programs currently being debugged.

CMPJRNIMG (Compare Journal Images) Command



Notes:

1 The second element (last-receiver) can be specified only if a value other than *CURRENT is specified for the first element (first-receiver).

P All parameters preceding this point can be specified in positional form.

Purpose

The Compare Journal Images (CMPJRNIMG) command allows the user to compare and note the differences between (1) the before and after images of record-level changes (updates, deletes, rollback-updates, and rollback-deletes) for a specific file and/or member, (IMAGES(*BOTH) must be specified for the Start Journal Physical File (STRJRNPF) command), or (2) the after and *previous after* image of a particular relative record (IMAGES(*AFTER) is specified for the STRJRNPF command). The output of the command is directed to a printer.

If before and after images are compared, the journaled changes can be compared for only one or all of the records in the specific file or member. The comparison can also be limited by a specific journal receiver range, or by a range of journal entries in a specific journal receiver range.

The printed output shows the record image before the change was made, followed by the record image after the change, followed by a line that indicates (with asterisks) the specific change in the record on a character-by-character basis, instead of on a field-by-field basis. If the journaled file has null-capable fields, the null value indicators that correspond to the before-image of the record are compared with the null value indicators that correspond to the after-image of the record. This is done on a field-by-field basis.

If there is no journal entry satisfying the search value specified, the command ends.

Restrictions:

1. The result of the comparison is sent only to the system printer.
2. The file/member specified must currently exist on the system and must have been journaled.
3. Only one member can be processed per command.

4. The comparison of journal images ends if one of the following conditions occurs:
   - The member was saved with storage freed.
   - The member was restored.
   - The member was cleared.
   - The member was initialized.
   - The member was reorganized.
   - The member was deleted.
   - The member was in use when the system ended abnormally.
   - Journaling the member was stopped.
   - The member had the changes applied or removed (by the Apply Journaled Changes (APYJRNCHG) command or the Remove Journaled Changes (RMVJRNCHG) command).
5. If the sequence number is reset in the range of receivers specified, the first occurrence of FROMENT or TOENT is used if the parameters are specified. The FROMENT and FROMTIME parameters are mutually exclusive, as are the TOENT and TOTIME parameters.

Required Parameter

FILE

Specifies the qualified name of the physical database file for which the journaled record-level changes are compared.

The name of the physical file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

physical-file-name: Specify the name of the physical file.

Optional Parameters

MBR

Specifies the name of the member in the file that has its journal entries compared.

***FIRST:** The first member in the database file is used.

member-name: Specify the name of the member for which record-level changes are compared.

RCVRNG

Specifies the first and last journal receivers used in the comparison. The system starts the comparison with the first journal receiver (specified by the first value) and proceeds through receivers until the last receiver (specified by the last value) is processed. If dual receivers are used at any time, the first of the dual receivers is always used when chaining through the receivers. If any problem (such as damaged receivers or receiver not found) occurs in the receiver chain before the comparison starts, the system tries to use the second of the dual receivers. If the second of the receivers is damaged or not found, or if a problem occurs during the operation, the comparison ends.

Note: If the maximum number of receivers in the range is exceeded (256), an exception is sent and no entries are compared.

***CURRENT:** The journal entries in the currently attached receiver are used in the comparison.

The name of the journal receiver can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

Element 1: First receiver

first-receiver: Specify the name of the first journal receiver that contains the journal entries that are compared.

***CURRENT:** The comparison of journal entries continues for all receivers in the chain which started with the receiver specified by the first parameter value through the currently attached journal receiver.

The name of the journal receiver can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

Element 2: Last receiver

last-receiver: Specify the name of the last journal receiver that contains the journal entries to be compared. If the end of the receiver chain is reached before finding a receiver of the specified name, an error message is sent, the command ends, and no entries are compared.

FROMENT

Specifies the first journal entry to be compared.

***FIRST:** The first journal entry in the journal receiver range specified is the first entry considered for the comparison.

starting-sequence-number: Specify the sequence number at which the comparison begins.

FROMTIME

Specifies the date and time of the first journal entry to be compared. The journal entry with the specified date and time or the next later journal entry is the starting point for the comparison. The format of the date must be defined by the job attributes DATFMT and, if separators are used, DATSEP. The time is specified in 24-hour format with or without a time separator as follows:

- With a time separator, specify a string of 5 or 8 digits where the time separator separates the hours, minutes, and seconds. If this command is entered from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command fails.
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds. Valid values for **hh** range from 00 through 23. Valid values for **mm** and **ss** range from 00 through 59.

Element 1: Date When Comparison Starts

starting-date: Specify the date on which comparison of the first entry starts.

Element 2: Time When Comparison Starts

starting-time: Specify the time at which comparison of the first entry starts.

TOENT

Specifies the last entry considered for the comparison.

***LAST:** The last journal entry in the journal receivers specified is the final entry compared.

ending-sequence-number: Specify the sequence number of the last journal entry to be compared.

Note: The values specified for the FROM and TO parameters can be the same (for example, FROMENT(234) and TOENT(234) can be specified).

TOTIME

Specifies the time and date of the last journal entry to be compared. The journal entry with the specified date and time or the latest earlier journal entry is the ending point for the comparison of journal entries. The format of the date must be defined by the job attributes DATFMT and, if separators are used, DATSEP. See the FROMTIME parameter for a description of time formats.

Element 1: Date When Comparison Ends

ending-date: Specify the date on which comparison ends.

Element 2: Time When Comparison Ends

ending-time: Specify the time at which comparison ends.

CMPOPT

Specifies the type of record images that are compared for record-level changes in the specified file. If CMPOPT(*AFTER) is specified, (1) the default value *ALL must be specified for the JOB, PGM and CMTCYCID parameters, and (2) a relative record number must be specified for the RCDNBR parameter.

***BOTH:** The before-images of the journal entries are compared with the after-images of the journal entries.

***AFTER:** The after-images of the record (specified in the RCDNBR parameter) are compared with previous after-images.

RCDNBR

Specifies the relative record number in the file for which the journal entry images are compared.

***ALL:** The journaled changes for all records in the physical member are compared.

relative-record-number: Specify the relative record number in the physical member for which images are compared. If a value is specified, only changes for the specified record are compared.

JOB

Specifies that the comparison is of journal entries for a particular job.

A job identifier is either the special value *ALL or a qualified name with up to three elements. For example:

```
*ALL
job-name
user-name/job-name
job-number/user-name/job-name
```

More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***ALL:** The comparison is not limited to entries for a particular job.

job-name: Specify the name of the job whose journaled changes are considered for comparison.

user-name: Specify the name of the user of the job whose journaled changes are considered for comparison.

job-number: Specify the number of the job whose journaled changes are considered for comparison.

PGM

Specifies that the comparison is of journal entries for a particular program.

***ALL:** The comparison is not limited to entries for a particular program.

program-name: Specify the name of the program whose record-level changes are to be considered for comparison. Only changes journaled for this program are considered for comparison.

USRPRF

Specifies that the comparison is of journal entries for a particular user profile name. The user profile name is the user profile under which the job is run that causes the entries to be journaled.

***ALL:** The comparison is not limited to entries for a particular user profile.

user-name: Specify the name of the user profile that has journaled changes to be compared. Only journaled changes for this user profile are to be considered for comparison.

CMTCYCID

Specifies that the comparison is of journal entries for a particular commit cycle.

***ALL:** The comparison is not limited to entries for a particular commit cycle.

commit-cycle-identifier: Specify the identifier for the commit cycle whose journaled changes are to be considered for comparison. Only journaled changes for this commit cycle are considered for comparison.

OUTFMT

Specifies the format in which the objects are shown.

***CHAR:** The record images are shown in character format.

***HEX:** Record images are shown in both character format and hexadecimal format.

Examples

Example 1: Comparing Before-Images with After-Images

```
CMPJRNIMG   FILE(QGPL/PF)
```

This command compares the journaled record-level changes for the first member of file PF in the QGPL library. The entries compared are in the currently-attached receiver, starting with the first entry and ending with the last entry. All entries with both before-images and after-images that satisfy the selection values are eligible to be compared. The before-images of the entries are compared with the after-images of the entries.

Example 2: Comparing After-Images with Previous After-Images

```
CMPJRNIMG   FILE(MYLIB/PAYROLL)  MBR(APRIL)
  RCVRNG((RCVLIB/RCV3) (*CURRENT))  FROMENT(200)
  TOENT(500)  CMPOPT(*AFTER)  RCDNBR(999)  OUTFMT(*HEX)
```

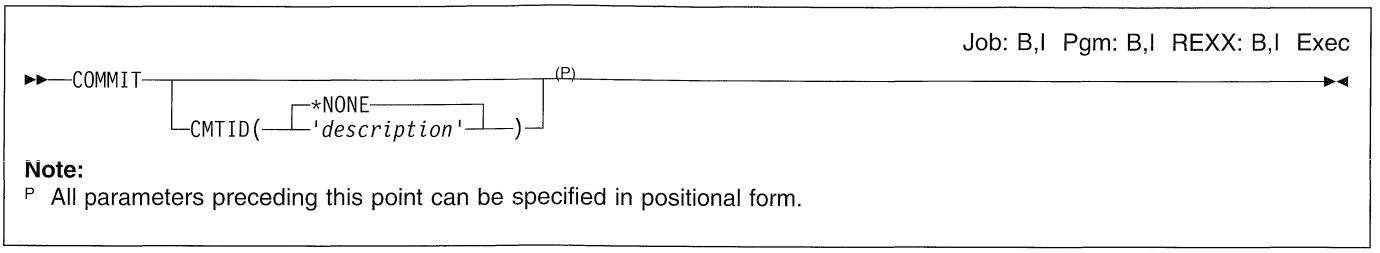
This command compares the journaled record-level changes for the member named APRIL in file PAYROLL in MYLIB, beginning with receiver RCV3 in RCVLIB and ending with the current receiver. The range of entries compared starts with entry 200 and ends with entry 500. Only the after-images and previous after-images are compared. The comparison is limited to record number 999. The output is printed in hexadecimal format.

Example 3: Specifying Journal Entry Date and Time

```
CMPJRNIMG   FILE(USERLIB/MYFILE)  MBR(*FIRST)
  RCVRNG((RCV2) (USERLIB/RCV5))
  FROMTIME('7/04/87' 120000)  TOENT(1000)
```

This command compares the journaled record-level changes for the first member of file MYFILE in USERLIB, beginning with receiver RCV2 in *LIBL and ending with receiver RCV5 in USERLIB. The date and time of the first journal entry to be compared is 7/4/87 12:00:00, and the ending record sequence number considered for the comparison is 1000.

COMMIT (Commit) Command



Note:
P All parameters preceding this point can be specified in positional form.

Purpose

The Commit (COMMIT) command is used to complete the current transaction and to establish a new commitment boundary for the commitment definition associated with the program issuing the command.

The Start Commitment Control (STRCMTCTL) command must be issued first to establish the commitment definition before the COMMIT command is issued; otherwise, a message is sent.

When the COMMIT command is issued, all pending changes made to resources under commitment control for the commitment definition since the last commitment boundary was established are made permanent. A commitment identifier can be specified that is associated with this set of changes. The commitment identifier is placed in the CM (changes committed) journal entry if any files have been opened since the commitment definition was started.

The commitment identifier is also used by the system when updating the notify object if it needs updating during activation group end, job end, or IPL (initial program load) processing.

No error occurs if there are no resources under commitment control for the commitment definition at the time the commit is issued. All record locks held for files opened under commitment control for the commitment definition are released when the commit is issued. Locks on object level commitment control resources, acquired when the resources are created or changed during the transaction are released when the commit is issued.

More information on commitment control is in the *Advanced Backup and Recovery Guide*.

Optional Parameter

CMTID

Specifies the text used to identify a group of changes committed with the commitment boundary.

***NONE:** No text is used to identify the group of changes committed with this commitment boundary.

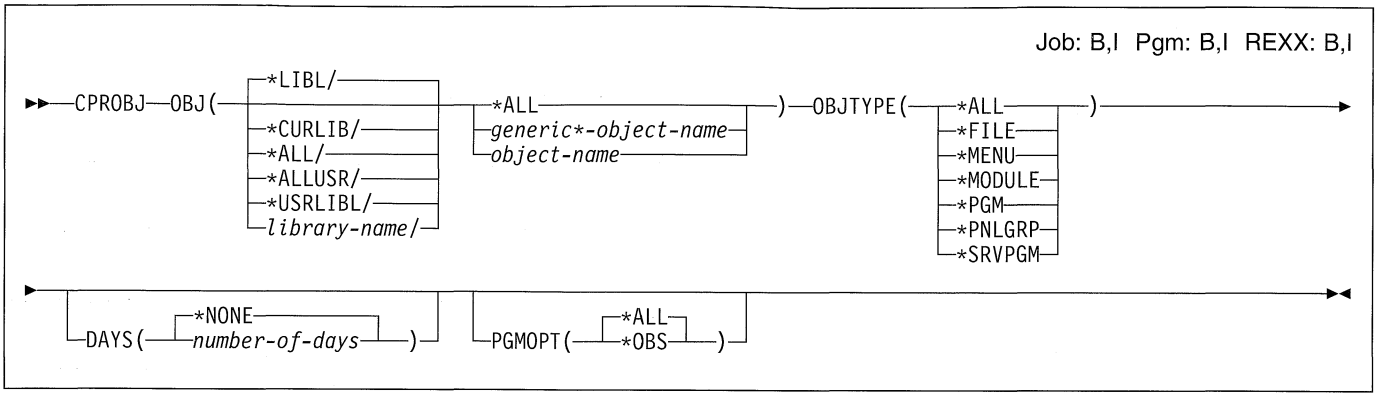
'description': Specify a maximum of 4000 characters, enclosed in apostrophes, to identify the group of changes being committed with this commitment boundary.

Example

```
COMMIT   CMTID('Account #123456 changes end')
```

This command specifies that all changes made to this point for the commitment definition associated with the program issuing the command are committed. The commitment identifier is specified and may be used by the system when updating the notify object if it needs updating during activation group end, job end, or IPL processing.

CPROBJ (Compress Object) Command



Purpose

The Compress Object (CPROBJ) command compresses programs, panel groups, menus, display files, printer files, modules, and service programs.

- *Compressed Objects* are objects that consume less storage space than decompressed objects. When a compressed object is used or a compressed program is called, a decompressed version of the object automatically becomes available to the user.
- *Decompressed Objects* are objects that use the system storage space allocated to them and are in a final, ready-to-use state.
- *Temporarily Decompressed Objects* are temporarily decompressed copies of compressed objects. The system allocates storage space for the temporary copies until the system or the user determines that the temporary storage space needs to be reclaimed.

  Temporary storage is automatically reclaimed when:

  - The RCLTMPSTG command is run
  - The next initial program load (IPL) is run
  - The object is used often enough to cause the system to permanently decompress it

  When an object is permanently decompressed, the compressed version of the object is destroyed as well as any temporary forms of the object; however, compressed versions remain intact as long as the objects are temporarily decompressed.

Restrictions:

1. The user must have object management authority to the object specified and *READ authority to the library containing the object.
2. Objects that were saved with storage freed cannot be compressed or decompressed.
3. Objects that are compressed cannot be saved for a release prior to Version 2 Release 1 of the OS/400 system.
4. Programs without a valid validation value are not compressed.
5. Programs that were created before Version 1 Release 3 of the OS/400 system and have not been retranslated (using the Change Program (CHGPGM) command) can not be compressed because no validation value has been generated.
6. To compress a system program, the user must end all active subsystems.
7. To prevent abnormal end of a program, the program must not be running in the system when it is compressed.

Required Parameters

OBJ

Specifies the qualified name of the object to be compressed.

The name of the object can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

***USRLIBL:** Only the libraries in the user portion of the job's library list are searched.

***ALL:** All libraries in the system portion of the job's library list, including QSYS, are searched.

***ALLUSR:** All user libraries are searched. All libraries with names that do not begin with the letter Q are searched except for the following:

```
#CGULIB    #DFULIB    #RPGLIB    #SEULIB
#COBLIB    #DSULIB    #SDALIB
```

Although the following Qxxx libraries are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are considered *user libraries*, and are also searched:

```
QDSNX      QPFRDATA   QUSER38
QGPL       QRCL       QUSRSYS
QGPL38     QS36F      QUSRVxRxMx
```

Note: A different library name, of the form QUSRVxRxMx, is added with each release. VxRxMx is the version, release, and modification level of the library.

library-name: Specify the name of the library to be searched.

***ALL:** All objects in the specified library, that can be compressed, are compressed.

generic-object-name:* Specify the generic name of the object. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be compressed only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

object-name: Specify the name of the object that is compressed.

OBJTYPE

Specifies the type of the object to be compressed.

***ALL:** All menus, panel groups, display and printer device files, programs, modules, and service programs with the specified object name are compressed.

***FILE:** The file with the specified object name is compressed (display and printer files only).

***MENU:** The menu with the specified object name is compressed.

***MODULE:** The module with the specified object name is compressed.

***PGM:** The program with the specified object name is compressed.

***PNLGRP:** The panel group with the specified object name is compressed.

***SRVPGM:** The service program with the specified object name is compressed.

Optional Parameters

DAYS

Specifies the number of days an object has not been used or changed. If the object has not been used or changed for more than the specified number of days, it is compressed. If it has been used or changed, it is left decompressed.

***NONE:** The object is compressed regardless of the number of days it has not been used or changed.

number-of-days: Specify the number of days an object must be unused for it to be compressed. Valid values range from 1 through 366 days.

PGMOPT

Specifies, for *PGM or *SRVPGM objects, the program option that indicates whether the entire program (instruction stream and observability tables) or only the observability tables are compressed.

***ALL:** The entire program or service program is compressed.

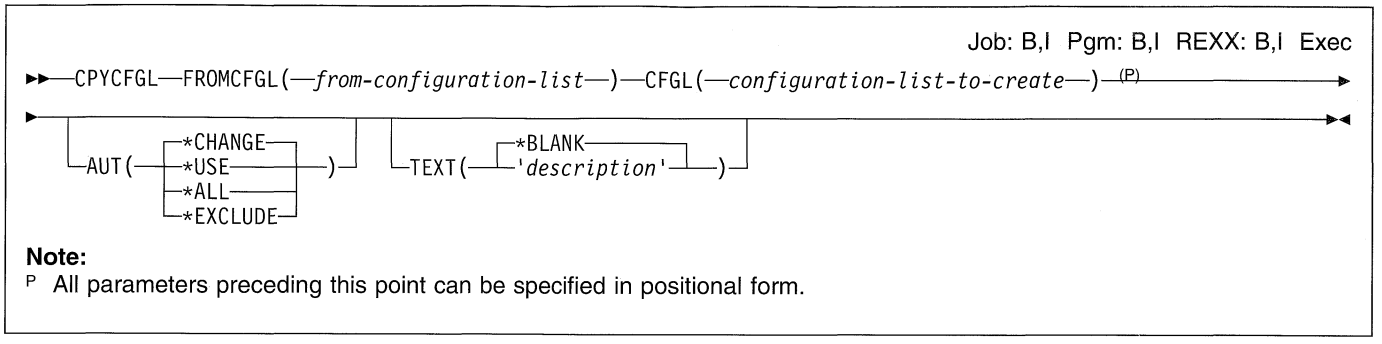
***OBS:** Only the observability tables of the program or service program are compressed.

Example

```
CPROBJ  OBJ(QGPL/*ALL)
  OBJTYPE(*FILE)
```

This command compresses all display and printer files of type *FILE in library QGPL.

CPYCFGL (Copy Configuration List) Command



Note:

(P) All parameters preceding this point can be specified in positional form.

Purpose

The Copy Configuration List (CPYCFGL) command creates a configuration list as a copy of an existing configuration list.

Required Parameters

FROMCFGL

Specifies the configuration list from which to copy. APPN local and remote location lists and asynchronous location lists are not allowed for copying. Only one of these list types is allowed per system.

CFGL

Specifies the name of the configuration list being created.

Optional Parameters

AUT

Specifies the authority given to users who do not have specific authority to the configuration list, who are not on an authorization list, and whose user group has no specific authority to the configuration list.

***CHANGE:** The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority.

***USE:** The user can perform basic operations on the configuration list, such as running a program or reading a file. The user cannot change the configuration list. *USE authority provides object operational authority and read authority. If the object is an authorization list, the user cannot add, change, or remove users.

***ALL:** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence and specify the security for the object, change the object, and perform basic functions on the object. The user can change ownership of the configuration list.

***EXCLUDE:** The user cannot access the configuration list.

TEXT

Specifies text that briefly describes the configuration list. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***BLANK:** Text is not specified.

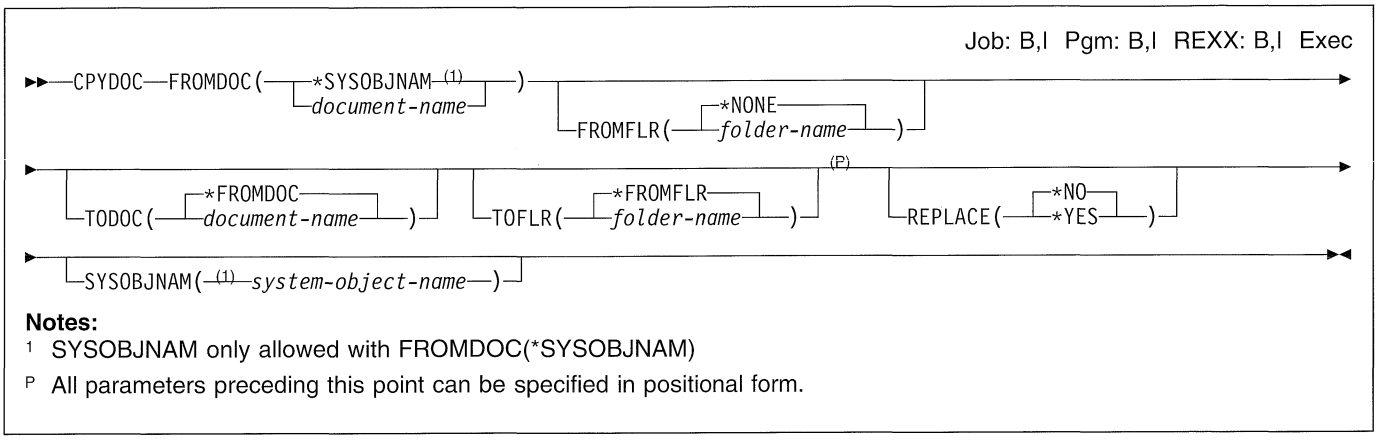
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CPYCFGL  FROMCFGL(CONFIG01)  CFGL(CONFIG02)
```

This command copies the configuration list named CONFIG01 to a new configuration list name CONFIG02.

CPYDOC (Copy Document) Command



Notes:
1 SYSOBJNAM only allowed with FROMDOC(*SYSOBJNAM)
P All parameters preceding this point can be specified in positional form.

Purpose

The Copy Document (CPYDOC) command allows the user to copy a document from a folder into another folder, to copy a document that is not in a folder into a folder, and copy a document into the same folder or another folder with a different name.

The document copy is not indexed, regardless of whether or not the original document is indexed. If the document copy already exists and is already indexed, the index entry will not match the new content of the document copy, as the document is not reindexed. If you want the document copy to be indexed or reindexed, use the Add Text Index Entry (ADDTXTIDXE) command after doing the copy.

Restrictions:

1. If the user is replacing a document, then the user must have *CHANGE authority for that document.
2. If the user is creating a new document, then the user must have *CHANGE authority for the folder that is to contain it. The new document is to have the same authorization as the document from which it is copied.
3. The user must have use authority for the from-document.

Required Parameter

FROMDOC
Specifies the name of the document being copied.

Note: If FROMDOC(*document-name*) is specified, a folder name must be specified on FROMFLR. If FROMDOC(*SYSOBJNAM) is specified, a system object name must be specified on SYSOBJNAM.

***SYSOBJNAM:** A system object name is used to identify the document being copied.

document-name: Specify the name of the document that is copied.

Optional Parameters

FROMFLR
Specifies the name of the folder that contains the document to copy.

***NONE:** A folder name is not specified for the document. FROMFLR(*NONE) must be specified if the document is not in a folder. FROMFLR(*NONE) cannot be specified if a document name is specified for FROMDOC.

folder-name: Specify the name of the folder from which the document is copied.

TODOC
Specifies the folder into which the output document is copied.

Note: The user cannot specify both TODOC(*FROMDOC) and TOFLR(*FROMFLR) to designate the copied document in its respective folder.

***FROMDOC:** The name of the new document is the same as the name specified on the FROMDOC parameter.

document-name: Specify the name of the new output document.

TOFLR
Specifies the folder into which the document is copied.

***FROMFLR:** The folder name is the same as that specified in the FROMFLR parameter and the document is copied into the same folder.

folder-name: Specify the name of the folder into which the document is copied.

REPLACE
Specifies whether the document specified by TODOC can be replaced. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***NO:** A new document, specified on the TODOC parameter, is created as a copy of the document being copied.

If a document already exists in the folder specified on the TOFLR parameter, no copy is made.

***YES:** The document replaces an existing document with the same name in the folder specified on the TOFLR parameter. If no document with the same name exists in the folder, a new document is created.

SYSOBJNAM

Specifies the system object name. This parameter is valid only when DLO(*SYSOBJNAM) or DOCL(*SYSOBJNAM) is specified. A full ten characters must be specified.

Examples

Example 1: Copying a Document

```
CPYDOC    FROMDOC(MYDOC)  FROMFLR(MYFLR)
  TODOC(MYDOC2)  TOFLR(MYFLR2)  REPLACE(*YES)
```

This command copies document MYDOC located in folder MYFLR to document MYDOC2 located in folder MYFLR2. If document MYDOC2 already exists in MYFLR2, the system replaces it with a copy of document MYDOC; otherwise, MYDOC2 is created in MYFLR2 as a copy of MYDOC in MYFLR.

Example 2: Copying Document and Keeping Source Document Name

```
CPYDOC  FROMDOC(*SYSOBJNAM)  SYSOBJNAM(AMBT133080)
  TODOC(MYDOC4)  TOFLR(MYFLR)
```

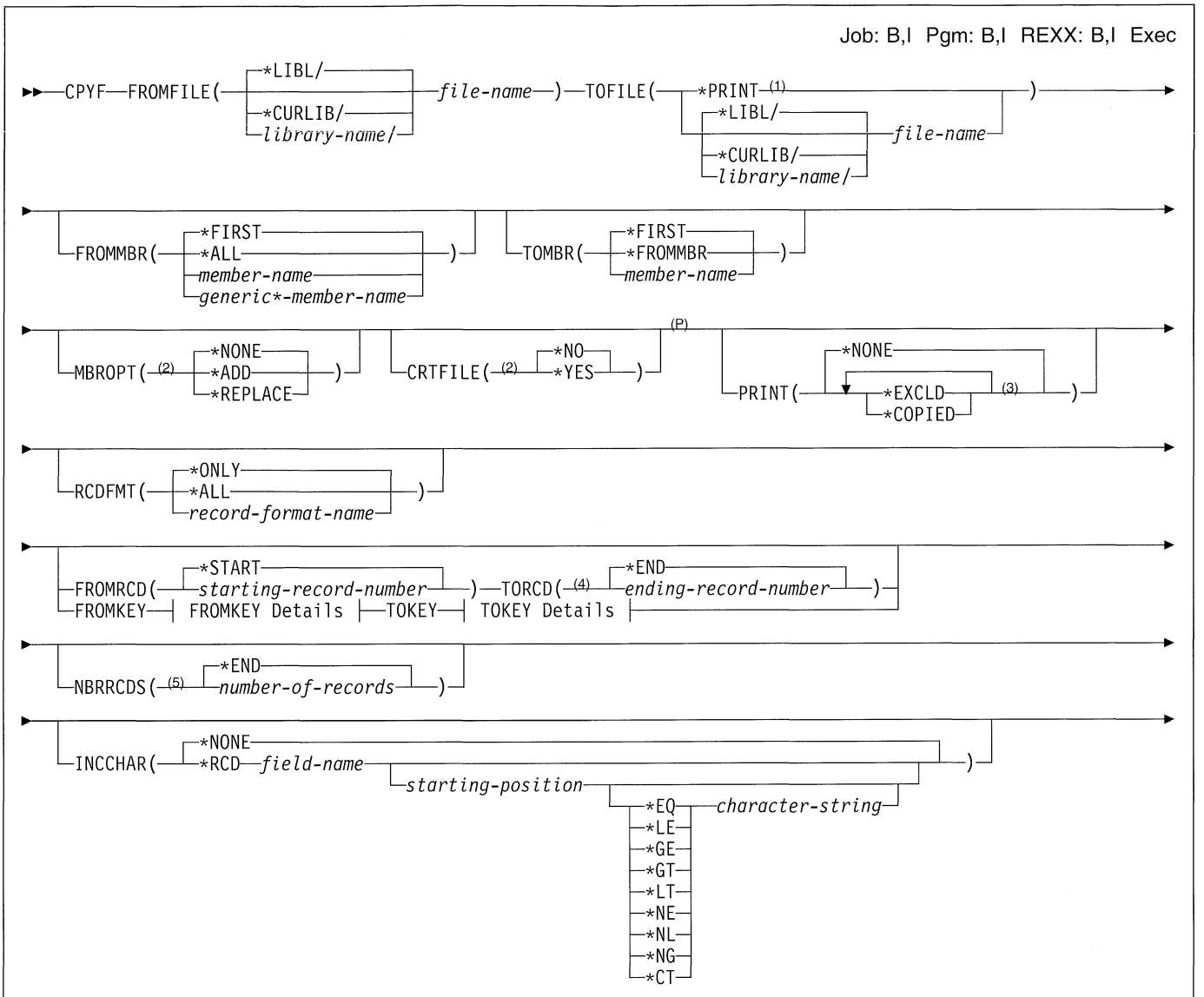
This command copies a document, identified by the system object name, to document MYDOC4 located in folder MYFLR. The document name will be the same as the name of the source document.

Example 3: Copying Document to Document in Same Folder

```
CPYDOC  FROMDOC(XYZ)  FROMFLR('MYFLR/TEST')
  TODOC(NEW)
```

This command copies document XYZ located in folder MYFLR/TEST to document NEW in the same folder. If document NEW already exists, an error message is sent.

CPYF (Copy File) Command



Notes:

1 If TOFILE(*PRINT) is specified, either *FIRST or *FROMMBR must be specified on theTOMBR parameter.

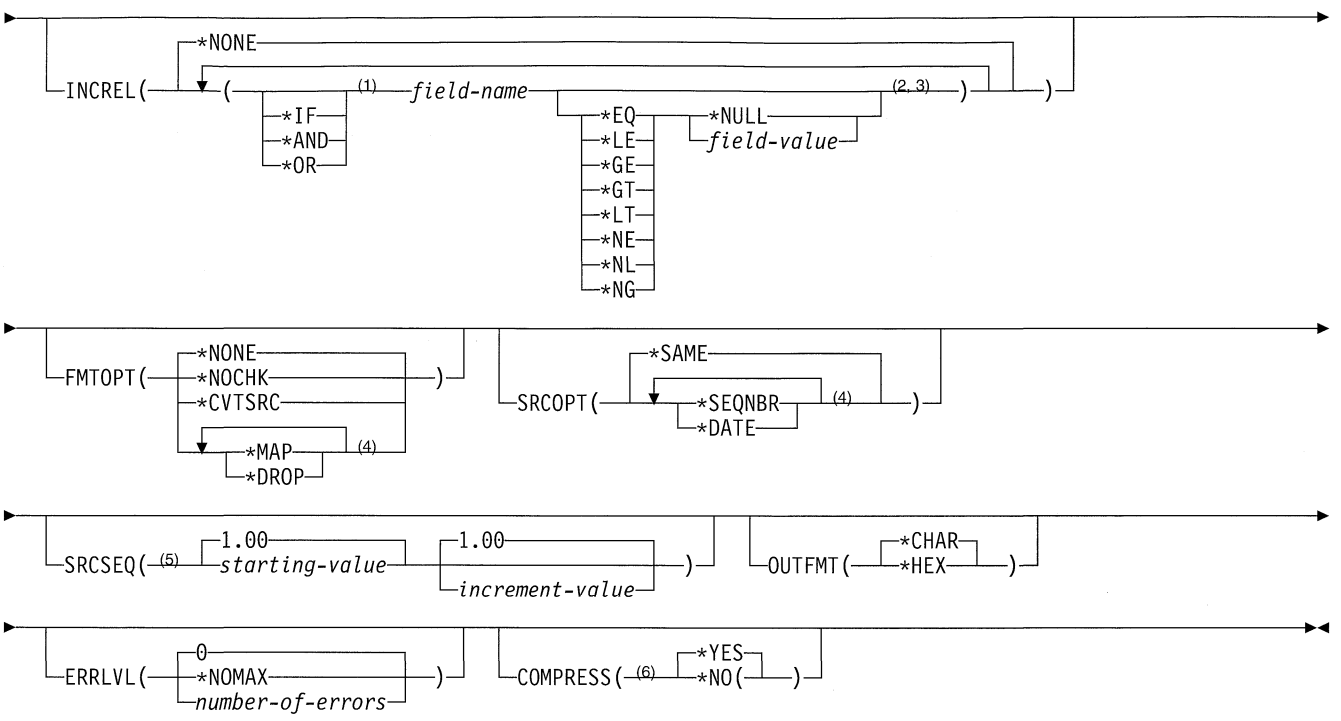
2 If CRTFILE(*YES) is specified and the to-file does not exist, a library-name (not *LIBL) must be specified for the TOFILE parameter and MBROPT(*ADD) is assumed. If the to-file is an existing physical file, the MBROPT value must be either *ADD or *REPLACE.

3 A maximum of 2 repetitions

4 If TORCD is specified, NBRRCDS cannot be specified.

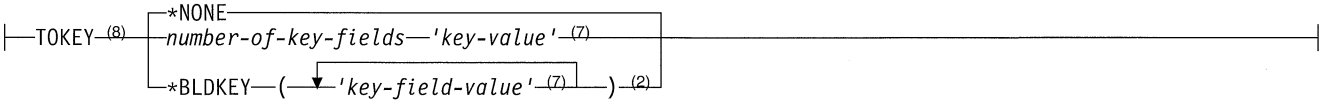
5 If NBRRCDS is specified, neither TORCD nor TOKEY can be specified.

P All parameters preceding this point can be specified in positional form.



FROMKEY Details:

TOKEY Details:



Notes:

1 *IF can be specified the first time only; either *AND or *OR must be specified when repeating selections.

2 A maximum of 50 repetitions

3 Only the operators *EQ and *NE are allowed if *NULL is specified.

4 A maximum of 2 repetitions

5 When SRCSEQ is specified, SRCOPT is required.

6 COMPRESS(*NO) is not allowed with TOFILE(*PRINT), and requires the default value for the INCCHAR and INCREL parameters. COMPRESS(*NO) is not allowed when copying from a delete capable file to a non-delete capable file.

7 A maximum of 256 characters can be specifed.

8 If TOKEY is specified, NBRRCDS cannot be specified.

Purpose

The Copy File (CPYF) command copies all or part of a database or external device file to a database or external device file. Copy operations can be done from physical, logical, diskette, tape, inline, or DDM files to physical, diskette, tape, printer, or DDM files.

Note: For more information on the use of DDM files, see the *DDM Guide*.

Some of the specific functions that can be performed by the CPYF command include the following:

- The CPYF command can copy one database file member, a set of members with a common generic name, or all members in a database file. Many database file members can be copied to any valid to-file type except to more than one label on tape (FROMMBR and TOMBR parameters).
- The CPYF command can copy one diskette file label identifier, a set of labels with a common generic name, or all labels from a diskette file. Many diskette labels

can be copied to any valid to-file type except to more than one label on tape (FROMMBR and TOMBR parameters).

- The CPYF command can add records to an existing physical file member or replace the contents of a receiving physical file member (MBROPT parameter).
- The CPYF command can select certain records to be copied using one of the following methods:
  - The CPYF command can select a particular format in a logical file that has more than one record format (RCDFMT parameter).
  - The CPYF command can limit the range of records copied based on starting and ending relative record numbers (FROMRCD and TORCD parameters).
  - The CPYF command can limit the range of records copied based on starting and ending record key values (FROMKEY and TOKEY parameters).
  - The CPYF command can limit the maximum number of records copied based on a number of records (NBRRCDS parameter).
  - The CPYF command can compare a specified character string to a position in a record or field, or scan the record or field for the character string (INCCHAR parameter).
  - The CPYF command can compare a specified value to one or more fields in each record. The record is selected based on a logical expression over the field comparisons (INCREL parameter).
- The CPYF command can copy records between database files whose from-file and to-file record formats are different, and convert records when the from-file and to-file are different file types (source and data, FMTOPT parameter).
- The CPYF command can insert new sequence numbers and a zero date in the sequence number and date fields of records copied to a source physical file (SRCOPT parameter).
- The CPYF command can select a printout of either character or hexadecimal format (OUTFMT parameter).

Several other commands offer copy functions with a more specific set of parameters that are tailored to the to-file or from-file. For more information, refer to the following command descriptions:

CPYFRMDKT (Copy from Diskette)

CPYFRMTAP (Copy from Tape)

CPYSRCF (Copy Source File)

CPYTODKT (Copy to Diskette)

CPYTOTAP (Copy to Tape)

CPYFRMQRYF (Copy from Query File)

Error Handling: The escape message CPF2817 is sent for many different error conditions that can occur during a copy operation. At least one diagnostic message that indicates the specific error condition always comes before the escape message. More information on handling errors is in the *Data Management Guide*.

Status Message: During the running of the CPYF command, message CPI2801 is sent as a status message informing the interactive user that a copy is occurring. More information on preventing status messages from appearing is in the *CL Programmer's Guide*.

Restriction: During the time a CPYF request is run, the file named on the TOFILE parameter may be locked (similar to an *EXCL lock with no timeout) so that no access is possible.

Table 20, at the end of this command description, shows all of the CPYF parameters and indicates for which file types each parameter is valid. The parameters that can be used with all the database and device files are: PRINT, NBRRCDS, and INCCHAR. The parameters are listed down the left side, and the file types (and if each can be a from-file and/or a to-file) are shown across the top. An X indicates that the associated parameter is valid for the type and use of file under which it occurs. No X is shown when the parameter is either invalid or ignored (does not apply).

Required Parameters

FROMFILE

Specifies the qualified name of the database, inline data file, or device file that contains the records being copied. A database file can be a physical or logical file. A device file can be a diskette file or tape file.

The name of the file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

file-name: Specify the name of the file that contains the records to be copied.

TOFILE

Specifies the qualified name of the file that receives the copied records.

Note: A device file can be a diskette file, tape file, or printer file. However: (1) If the from-file and to-file are both diskette files, the to-file must be spooled (SPOOL(*YES) must be specified on the Create Diskette File (CRTDKTF), Change Diskette File (CHGDKTF), or Override Diskette File (OVRDKTF) command). (2) An externally described printer file cannot be specified.

If the device file is a printer file or if TOFILE(*PRINT) is specified, shift-out shift-in (SO-SI) character pairs are not added around the graphic data. OUTFMT(*HEX) can be specified to print the data in hexadecimal format.

***PRINT:** The data is copied to the IBM-supplied printer device file QSYSPRT and the file is formatted according to the OUTFMT parameter.

The IBM-supplied printer file QSYSPRT may not be overridden to a different file name, and it must have the RPLUNPRT(*YES) and CTLCHAR(*NONE) attributes.

The name of the file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

file-name: Specify the qualified name of the physical file or device file that receives the copied records. If no library qualifier is specified, *LIBL is used to locate the file. However, if CRTFILE(*YES) is specified and the specified file cannot be found, the file name must be qualified with a library name. When the physical to-file is created, it is placed in the specified library.

Optional Parameters

FROMMBR

Specifies the database file member name, or the diskette or tape file label identifier in the from-file, that is copied. A generic name or *ALL can be specified to copy many database members or diskette file labels, but only a single file label identifier can be copied if the from-file is a tape file. If the from-file is a spooled inline file, FROMMBR(*FIRST) is required.

***FIRST:** The first member (in the order of creation date) in a database from-file is copied. If the from-file is a diskette or tape file, no label identifier is specified by the copy operation when the file is opened. For diskette, a label identifier (LABEL parameter) must be specified in the device file or on an OVRDKTF command. FROMMBR(*FIRST) is required if the from-file is an inline file.

***ALL:** All members of a database from-file, or all file label identifiers for a diskette from-file are copied. FROMMBR(*ALL) is not valid if the from-file is a tape file or inline data file.

If the to-file is a diskette or physical file, the data is copied to like-named to-file members or labels (if TOMBR(*FROMMBR) is specified), or all the from-file members/labels are copied to a single to-file member, diskette label or tape label. If the to-file is a printer file, each member or label is copied to a separate spooled file. If TOFILE(*PRINT) is specified, all the records are copied to a single print output file, and the records for each member or label identifier copied starts on a new print page.

If one of the files copied from a diskette is continued onto another volume, all the files on the continuation volume are also copied (or checked whether they should be copied if a generic name is specified).

member-name: Specify the name of the database from-file member or diskette or tape from-file label identifier of the file member being copied.

generic-member-name:* Specify a generic name to copy all database members that have names with the same prefix, or all diskette data files with the same prefix label identifier. Refer to the description of FROMMBR(*ALL) for more information about copying many from-file members or label identifiers. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be copied only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

TOMBR

Specifies the name of the file member that receives the copied records.

Note: If the to-file is a printer file, the TOMBR value must be either *FIRST or *FROMMBR. If a member name or TOMBR(*FROMMBR) is specified on the CPYF command, or if a to-member name is specified on an override and the member does not exist in the physical to-file, a member is added to the file by the copy operation.

This parameter value is valid for both a single from-file member or label, and for many from-file members or labels (specified by *ALL or a generic name for the FROMMBR parameter).

Specifying the *FIRST value on the TOMBR parameter is not allowed if the to-file is a physical file that has no members unless either a member name is specified in an override, or CRTFILE(*YES) is specified and the to-file does not already exist. When a physical file is created by the copy operation for the to-file and TOMBR(*FIRST) is specified, the from-file file name is used as the member name in the created file.

***FIRST:** The first member in the database to-file receives the copied records.

***FROMMBR:** Corresponding from-file and to-file member names or label identifiers are used. This parameter value is valid for both a single from-file member or label, and for many from-file members or labels (specified by *ALL or a generic name for the FROMMBR parameter). It is ignored if the to-file is a printer or if TOFILE(*PRINT) is specified. If the to-file is a diskette or tape file, or if the to-file is a database file and there is no member override, the TOMBR(*FROMMBR) value is valid only if the from-file is a database, diskette, or tape file.

member-name: Specify the name of the physical to-file member, or diskette or tape to-file label identifier, to receive the copied records. This parameter value is valid for both a single from-file member or label, and for many from-file members or labels (specified by *ALL or a generic name for the FROMMBR parameter). If a member with the specified name does not already exist in the physical to-file, the copy operation attempts to add a member with the specified name to the file.

MBROPT

Specifies whether the new records replace or are added to the existing records.

Note: If the records are being copied to an existing physical file, this parameter must specify either *ADD or *REPLACE. If the to-file does not exist but CRTFILE(*YES) is specified, the copy operation assumes MBROPT(*ADD) for all records copied to the file after it is created, regardless of the value specified on this parameter.

If *ADD is specified and the from-file is empty (contains no records), the copy operation completes normally. If *REPLACE is specified and the from-file is empty, the copy operation ends abnormally.

***NONE:** This parameter does not apply to this copy operation. When the to-file is an existing physical file, MBROPT(*NONE) is not allowed; either *ADD or *REPLACE must be specified to indicate whether records should be added or replaced in each to-file member that is used.

***ADD:** The system adds the new records to the end of the existing records.

***REPLACE:** The system clears the existing member and adds the new records.

CRTFILE

Specifies, when the CPYF command is used to copy from a physical file or logical file, whether a physical file is created to receive the data if the to-file does not exist. If the to-file already exists when the CPYF command is started, this parameter is ignored. If the to-file is created, the text and public authority of the from-file are used.

***NO:** The to-file must exist when the CPYF command is started. A physical file is not created to receive the data.

***YES:** If the to-file does not exist, a physical file is created with the name specified on the TOFILE parameter. In addition to the normal copy operation validity checks, the following special conditions must all be true for the copy operation to create a to-file:

- The from-file must be either a physical or logical file.
- A library name must be specified on the TOFILE parameter. The default value, *LIBL, is not allowed.
- There cannot be an override to a different file or library name. The values specified on the CPYF command for the to-file must be used.
- The user running the CPYF command must be authorized to add the file to the TOFILE library, and must have operational authority to the CRTPF command.
- A single record format must be used in the from-file. If the from-file is a logical file with multiple formats, the RCDFMT parameter must specify a record format name.

The members added to the physical file created by the copy operation have the names specified by the TOMBR parameter. If TOMBR(*FIRST) is specified, the to-file member has the same name as the from-file. The MBROPT parameter value is ignored when the to-file is created, and records are added to the new file members.

PRINT

Specifies whether copied records, excluded records, or both, are printed. The records are formatted according to the OUTFMT parameter value and written to the IBM-supplied printer file QSYSPRT. QSYSPRT must be spooled (SPOOL(*YES) must be specified in the device file or on the OVRPRTF command) when both copied and excluded records are printed, or when either copied or excluded printouts are requested with TOFILE(*PRINT), because separate printer files are opened for the printouts. If many from-file members or labels are copied, all the members are included in the printer files for copied or excluded records, and each member or label begins on a new print page.

If a selected range of records are requested to be copied using the FROMRCD, TORCD, FROMKEY, TOKEY, or NBRRCDS parameters, then only the records copied or excluded from that range are listed.

If the device file is a printer file, or if PRINT(*EXCLD) or PRINT(*COPIED) is specified, SO-SI characters are not added around the graphic data. OUTFMT(*HEX) can be specified to print the data in hexadecimal format.

***NONE:** Neither copied nor excluded records are printed.

***EXCLD:** Records excluded from the copy operation by the INCCHAR and INCREL parameters are printed.

***COPIED:** Copied records are printed.

RCDFMT

Specifies, for a copy operation from a database file only, the name of the record format that is copied. If the from-file is not a logical or physical file, RCDFMT(*ONLY) must be specified.

***ONLY:** The only record format in the from-file is copied. This value is required when the from-file is not a logical or physical file. When the from-file is a logical file, this value is valid only if the file has a single record format.

***ALL:** All record formats in the logical from-file are used. This value is valid for a physical file, and is also valid for a logical file even if the file has only a single record format. If the logical file has many formats, RCDFMT(*ALL) is allowed only if the to-file is a device file or *PRINT, or if the to-file is a physical file and the FMTOPT parameter value is specified as either *NOCHK or *CVTSRC.

record-format-name: Specify the name of the record format copied when the from-file is a logical or physical file. A record format name is optional if the logical file has only a single record format, but either a format name or *ALL must be specified if the from-file has more than one record format. If the logical file is based on more than one physical file member, only the data from the physical file members that are used to derive the specified record format is copied.

FROMRCD

Specifies the record number of the first record in the from-file (or each from-file member) copied. A FROMRCD record number is not valid if a value other than *NONE is specified for the FROMKEY or TOKEY parameters, and is not allowed if the from-file is a keyed logical file.

If the from-file is a physical file or a logical file with an arrival sequence access path, the FROMRCD value is a relative record number that counts both the deleted and nondeleted records that precede it. If the from-file is a device file or inline file, the FROMRCD value is a record number that includes only nondeleted records (even for an I-format diskette file).

If COMPRESS(*YES) is used, and the specified record in a database file member is deleted, the copy operation starts with the first nondeleted record (if any) after the specified record number.

***START:** The copy operation begins with the first record in the file, as determined by the access path or with the first record determined by the FROMKEY parameter value.

starting-record-number: Specify a record number, ranging from 1 through 2147483646, that identifies the first record copied from the from-file. If both FROMRCD and TORCD record number values are specified, the FROMRCD value must be less than or equal to the TORCD value.

TORCD

Specifies the record number of the last record in the from-file (or each from-file member) that is copied. A TORCD record number is not valid if a value other than *NONE is specified for the FROMKEY or TOKEY parameters, or if a value other than *END is specified for the NBRRCDS parameter, or if the from-file is a keyed logical file.

***END:** Records are copied until the end-of-file condition is indicated by the from-file, or until the amount is larger than the TOKEY record key value or the NBRRCDS maximum number of records is reached.

ending-record-number: Specify a record number, ranging from 1 through 2147483646, that identifies the last record copied from the from-file. If both FROMRCD and TORCD record number values are specified, the FROMRCD value must be less than or equal to the TORCD value. If an end-of-file condition is reached before this record number is found, no error messages are issued.

FROMKEY

Specifies, when a file with a keyed access path is copied, that the key value of the first record in the from-file (or each from-file member) is copied. This parameter is valid only when a from-file is a keyed database file, and is not allowed if record number values are specified for the FROMRCD or TORCD parameters.

If no record in the from-file member has a key that is a match with the FROMKEY value, but there is at least one record with a key greater than the specified value, the first record copied is the first record with a key greater than the FROMKEY value. If the specified key value is greater than any record in the member, an error message is sent and the member is not copied.

***NONE:** The first record copied is not selected by key.

Keys With Single-Character String Values:

Specify both of the following elements to identify the key value of the first record copied from the from-file.

Element 1: Number Of Key Fields

number-of-key-fields: Specify the number of key fields used to locate the first record copied. This value must be a number less than or equal to the total number of key fields specified in the data description specification (DDS) for the from-file. If the number is less than the total number of fields in the key, a partial key is used to locate the first record copied.

Element 2: Key String Value

'key-value': Specify a character string (up to 256 characters) that specifies the actual key value for the number of key fields specified by Element 1. The key string value specified must be as long as the total length of all the key fields specified by Element 1, or undesirable results may occur. If the key string specified is shorter than required for the number of key fields used, the string must be padded on the right with zeros. The key

string value must be enclosed in apostrophes if it contains blanks or special characters, and it may be specified in hexadecimal format, which is useful if the key contains packed decimal or binary numeric fields, or is a variable-length character field. CCSID conversions are not performed on character fields when a single string is specified.

Keys With A List Of Values:

Specify both of the following elements to identify values for the individual fields of the key, which can be a composite key. This method is generally easier to use if the key contains numeric fields.

Element 1: Build Keys For A List Of Values

***BLDKEY:** Indicates that a list of values is provided for key fields (as opposed to a single-character string value for all fields in the key). *BLDKEY is not valid if any value in the list corresponds to a null-capable field.

Element 2: Key Field Value List

'key-field-value': Specify the list of values (up to 256 characters each) that is applied (in order) to corresponding fields in the from-file key. The maximum number of values (up to 50) allowed in the list is limited to the number of key fields defined in the data description specifications (DDS) for the from-file. If fewer values are provided than the total number of key fields defined for the file, a partial key is used to locate the first record copied.

The values are converted from the displayed form to the type defined in the key field definition. Values specified for character fields are converted from the current job CCSID to the key field CCSID. When a graphic field is specified, the input string (DBCS data) must be enclosed in SO-SI characters. The SO-SI characters are removed from the input string and the remaining DBCS data is converted from the associated DBCS CCSID of the current job to the DBCS CCSID of the graphic field. If a value is specified for a character field that is shorter than the actual key field, the value is padded on the right with blanks. A value specified for a numeric key field is converted to the type and precision defined in the DDS for the key field. If a value is either too large for the corresponding key field or cannot be converted to the type required for the key field, an error message is sent, and the copy operation is not done. For date, time, or timestamp fields, corresponding input values are converted to the format and separator form of the from-file field. For variable-length fields, only specify the character data, not the 2-byte length portion.

TOKEY

Specifies, when a file with a keyed access path is copied, the key value of the last record in the from-file (or each from-file member) copied. This parameter is valid only for a from-file that is a keyed database file, and is not allowed if record number values are specified for the FROMRCD or TORCD parameters, or if a number of records is specified for the NBRRCDS parameter.

If there is more than one record in a from-file member with a key that matches the TOKEY value, all those records are copied. If no record in the from-file member has a key that is a match with the TOKEY value, the last record copied is the last one (if any) with a key value less than that of the specified key value.

If there are both ascending and descending fields in the file key, the first (the far left) key field determines whether the copy operation uses an ascending or descending key test to look for the last record to copy.

The user must specify one of the following:

- ***NONE**
- Both elements of **Keys With Single-Character String Values**
- Both elements of **Keys With A List Of Values**

***NONE:** The last record copied is not selected by key.

Keys With Single-Character String Values

Specify the two values that identify the key value of the last record copied from the from-file.

Element 1: Number Of Key Fields

number-of-key-fields: Specify the number of key fields used to locate the last record copied. This value must be a number less than or equal to the total number of key fields specified in the data description specification (DDS) for the from-file. If the number is less than the total number of fields in the key, a partial key is used to locate the first record copied.

Element 2: Key String Value

'key-value': Specify a character string (up to 256 characters) that specifies the actual key value for the number of key fields specified by Element 1. The key string value specified must be as long as the total length of all the key fields specified by Element 1, or undesirable results may occur. If the key string specified is shorter than required for the number of key fields used, the string must be padded on the right with zeros. The key string value must be enclosed in apostrophes if it contains blanks or special characters, and it may be specified in hexadecimal format, which is useful if the key contains packed decimal or binary numeric fields, or is a variable-length character field. CCSID conversions are not performed on character fields when a single string is specified.

Keys With A List Of Values

Specify values for the individual fields of the key, which can be a composite key. This method is generally easier to use if the key contains numeric fields.

Element 1: Build Keys For A List Of Values

***BLDKEY:** Indicates that a list of values is provided for key fields (as opposed to a single-character string value

for all fields in the key). *BLDKEY is not valid if any value in the list corresponds to a null-capable field.

Element 2: Key Field Value List

'key-field-value': The list of values (up to 256 characters each) specified is applied (in order) to corresponding fields in the from-file key. The maximum number of values (up to 50) allowed in the list is limited to the number of key fields defined in the DDS for the from-file. If fewer values are provided than the total number of key fields defined for the file, a partial key is used to identify the last record copied.

The values are converted from the displayed form to the type defined in the key field definition. Values specified for character fields are converted from the current job CCSID to the key field CCSID. When a graphic field is specified, the input string (DBCS data) must be enclosed in SO-SI characters. The SO-SI characters are removed from the input string and the remaining DBCS data is converted from the associated DBCS CCSID of the current job to the DBCS CCSID of the graphic field. If a value is specified for a character field that is shorter than the actual key field, the value is padded on the right with blanks. A value specified for a numeric key field is converted to the type and precision defined in the DDS for the key field. If a value is either too large for the corresponding key field or cannot be converted to the type required for the key field, an error message is sent, and the copy operation is not done. For date, time, or timestamp fields, corresponding input values are converted to the format and separator form of the from-file field. For variable-length fields, only specify the character data, not the 2-byte length portion.

NBRRCDS

Specifies the maximum number of records in the from-file (or each from-file member) copied to the to-file. The records copied start either at the start of the file access path or at the record indicated by the value specified for the FROMRCD or FROMKEY parameter. The TORCD or TOKEY parameters can be used only if NBRRCDS(*END) is specified.

This parameter controls the number of records that are copied after the selection value (INCCHAR/INCREL) is applied. It does not control the number of records that are read.

***END:** Records are copied until the end-of-file condition is indicated for the from-file, unless either the TOKEY or TORCD parameter has been specified.

number-of-records: Specify the number of records, ranging from 1 to 2147483647, that are copied to the to-file. Fewer records are copied if an end-of-file condition occurs before the specified number of records have been copied.

INCCHAR

Specifies that records are copied or excluded based on the result of a comparison with a character string value and the data in some position of either a field in the record or the entire record. The comparison done can include searching the record or field for the specified character string value. If INCCHAR is specified for a logical file with many record formats and RCDFMT(*ALL) is specified, the character string is used for selecting records from all the formats.

If both the INCCHAR and INCREL parameters are specified, a record is copied only if it satisfies the requirements for both parameters.

***NONE:** No character string value comparison is used to select which records are copied.

Comparison Values: To specify the comparison that determines which records are copied, four values must be entered. Either *RCD or the name of a field must be entered, followed by the three values that control the comparison: starting position, operator, and character string value. All records that satisfy the relationship specified by the four values are copied to the to-file.

***RCD:** The character string value is compared with the data at the specified starting position in each record copied from the from-file.

Element 1: Record Format Field Name

field-name: Specify the name of a field in the record format that is used to make the comparison. The field must be defined as a character field in the data description specification (DDS) for the from-file. When the from-file is a device or inline file, or when the copy operation must process many record formats for a logical from-file (when RCDFMT(*ALL) is specified), a field name cannot be specified (but *RCD is allowed).

Element 2: Field Record Starting Position

starting-position: Specify the position in the field or record where the comparison starts. When a variable-length field name is specified, the position is in the data portion of the variable-length field. For graphic fields, the position is the DBCS character position. For any operator except *CT, the comparison is done for the length of the character-string value (up to 256 maximum) specified on Element 4 of this parameter. For the *CT operator, the field or record is scanned from the specified starting position to the end of the field or record to determine whether it contains the specified character string. The character string length plus the starting position must not be larger than the length of the field (when a field name is specified) or a record (when *RCD is specified).

Element 3: Operator Value

operator: Specify the operator that indicates the relationship that must exist between the specified portion of the record or field and the character string specified as the last part of the INCCHAR parameter for the record copied to the to-file. The operators that are used are:

*EQ Equal
*GT Greater than

*LT Less than
*NE Not equal
*GE Greater than or equal
*NL Not less than
*LE Less than or equal
*NG Not greater than
*CT Contains

Element 4: Character String Value

character-string: Specify the character-string (up to 256 characters in length) to be compared with the specified field or record. The character-string length plus the starting position must not be larger than the length of the field (when a field name is specified) or a record (when *RCD is specified).

The character-string value must be enclosed in apostrophes if it contains blanks or special characters, and it can be specified in hexadecimal format. If a field name is specified, the character-string value is converted from the current job CCSID to the field CCSID prior to running the comparison. For variable-length fields, specify only the character data to be compared, not the 2-byte length portion. If a field name is specified, any comparison to a field value that is the null value will test false. For graphic fields, specify the input string (DBCS data) within SO-SI characters. The SO-SI characters are removed from the input string and the remaining DBCS data is converted from the associated DBCS CCSID of the current job to the DBCS CCSID of the graphic field.

INCREL

Specifies that records are copied or excluded based on whether certain fields in the record contain data that satisfies specified relationships. As many as 50 field value relations are used to determine whether each record is copied. Include-relationship values are specified only for the INCREL parameter when the from-file is a physical or logical file, and are not valid for a copy operation from all record formats of a logical file with many formats (when RCDFMT(*ALL) is specified).

If both the INCCHAR and INCREL parameters are specified, a record is copied only if it satisfies the requirements for both parameters.

***NONE:** No field value relationships are used to select which records are copied.

Relationship Values: To specify the conditions under which records are copied, a set of values is specified for each condition. Each set must contain exactly four values:

1. One of the logical operators *IF, *AND, or *OR
2. The name of the field compared
3. One of the relational operators (from the list that follows)
4. The comparison value

Values 2 and 4 are compared for the relationship specified by value 3.

The value *IF must be specified as the first value in the first set of comparison values, if there is only one set or several sets of comparison values. If more than one set of comparison values are specified, either *AND or *OR must be specified as the first value in each set after the first set of values. Each INCREL relational set must be enclosed in parentheses.

***IF:** Identifies the first field value relationship that must be satisfied before a record is copied.

***AND:** The field value relational groups on both sides of the *AND value must all be satisfied before a record is copied.

***OR:** If the field value relational group on either side of the *OR value is satisfied, the record is copied.

Element 1: Field Name

field-name: Specify the name of the field compared. The field must exist in the from-file record format, and may be defined as either character or numeric in the DDS for the file.

Element 2: Operator Value

operator: Specify the operator that indicates the relationship that must exist between the field contents in the record and the field value specified as the fourth part of this INCREL relation for this relation to be true. The operators that are used are:

*EQ Equal
*GT Greater than
*LT Less than
*NE Not equal
*GE Greater than or equal
*NL Not less than
*LE Less than or equal
*NG Not greater than

Element 3: Field Value

***NULL:** *NULL can be used as the value to test whether the field value in a record is null. Only the operators *EQ and *NE are allowed if *NULL is specified. An "*EQ *NULL" relation is true only if a field value in a record is null. An "*NE *NULL" relation is true only if a field value in a record is not null.

field-value: Specify the value (up to 256 characters) to be compared with the contents of the specified field. The specified value cannot be another field name. The field value must be enclosed in apostrophes if it contains blanks or special characters, and it may be specified in hexadecimal format. If a CL variable is specified for the value, it must be a character variable. Any non-*NULL comparison to a field value in a record that is null will test false, regardless of the operator used. For variable-length fields, specify only the data portion of the value, not the 2-byte length portion.

Each field value specified is converted from the displayed format to the type defined by the field in the from-file record format. If a value is specified for a character

field that is shorter than the actual field, the comparison is performed using only the length of the character string value. A value specified for a character field is converted from the current job CCSID to the CCSID of the from-file field. A value specified for a numeric field is converted to an internal form with the same number of decimal numbers defined in the DDS for the field. For graphic fields, specify the input string (DBCS data) within SO-SI characters. The SO-SI pair is removed from the input string and the remaining DBCS data is converted from the associated DBCS CCSID of the current job to the DBCS CCSID of the graphic field. If a value is either too large for the corresponding record format field definition, or cannot be converted to the type required for the field, an error message is sent and the copy operation is not done.

FMTOPT

Specifies, when a physical or logical from-file is copied to a physical to-file, what field-level record format processing (if any) is done. This parameter is ignored if the from-file or to-file is a device or inline file, or if TOFILE(*PRINT) is specified. When either the from-file or to-file is not a database file, records are copied without any field checking and are truncated or padded with blanks or zeros, depending on the characteristics of the to-file.

Note: Additional information and examples of mapping, truncation, and padding of fields is in the *Data Management Guide* and the *Guide to Programming for Printing*.

When either FMTOPT(*CVTSRC) or FMTOPT(*NOCHK) is specified and the record data copied from any from-file record is not long enough to fill a to-file record, the extra bytes in the to-file record are set to a default value. If a default value was specified in the DDS (DFT keyword) for a field, that field is initialized to the specified default; otherwise, all numeric fields are initialized to zeros, and all character fields are initialized to blanks.

When field-level mapping, dropping, or both are done, any field in the to-file record format that is not set by a corresponding from-file field value (including mapping conversion errors) is set to the default value that was specified on the DFT parameter in the DDS for the file, or to a default value that depends on the field type: blanks for character fields, zeros for numeric fields, current date/time for time/date fields, and null values for null-capable fields.

***NONE:** No field mapping or dropping is done during the copy operation. This value is valid only if the from-file and to-file are not both database files, or if they are both database files and have the same record format. The record formats are the same only if every field exists in both the from-file and to-file formats, and has the same starting position and attributes in both formats. Attributes include whether or not a field is null-capable, CCSID, and the date/time format and separator (if the field is a date/time field). Null values are copied if *NONE is valid and both files are database files.

***NOCHK:** If the record formats of the database files are different, the copy operation continues despite the differences. Record data is copied directly (left to right) from one file to the other. FMTOPT(*NOCHK) is required when copying all record formats from a logical file with multiple formats (when RCDFMT(*ALL) is specified) to a physical file that is of the same type (source or data) as the from-file. If this value is specified, null values are ignored and no conversion of date/time data occurs and no CCSID conversions are done.

***CVTSRC:** This value is used to copy between database files, from a source file to a data file, or from a data file to a source file. It is valid only when the from-file and to-file are different types (source and data). The file type conversion is done as follows:

- If the to-file is a data file, the from-file sequence number and date fields are dropped, and the source data part of each from-file record is copied to the to-file, as described for FMTOPT(*NOCHK).
- If the to-file is a source file, sequence number and date fields are appended, and the from-file record data is copied to the source data part of each to-file record, as described for FMTOPT(*NOCHK). The SRCOPT and SRCSEQ parameters are used to control the sequence numbers created in the to-file. Null values are ignored and no conversion of date/time data occurs.

Note: When either the from-file or the to-file is not a database file, FMTOPT(*CVTSRC) is not required for copying from a source file to a data file or from a data file to a source file. Sequence number and date fields are appended or dropped automatically, depending on the file types. If the to-file is a source physical file, the SRCOPT and SRCSEQ parameters can be used to control the sequence numbers created for records copied to the to-file.

***MAP:** Fields with the same name in the from-file and to-file record formats are copied, and any fields in the to-file that do not exist in the from-file format are set to one of the following:

- The default value indicated by the DFT value in the data description specification (DDS) for the to-file.
- Blanks for character fields and zeros for numeric fields.
- Current date/time for date/time fields.
- The null value for null-capable fields.

If *MAP is specified, *DROP can also be specified. Mapped fields may have different starting positions in the from-file and to-file record formats.

If *MAP is specified and a valid conversion is defined between the from-file field CCSID and the to-file field

CCSID, the character data is converted to the to-file field CCSID.

If the from-file field CCSID or the to-file field CCSID is 65535, the character data is not converted.

*MAP allows conversion of date/time data and the copying of null values.

***DROP:** This value must be specified for field-level mapping if any of the field names in the from-file record format do not exist in the to-file format. If *DROP is specified, *MAP can also be specified. When *DROP is specified, all the field names that exist in both record formats must have the same attributes and relative positions in the from-file and to-file record formats, or *MAP must also be specified. Null values are copied if *DROP is valid.

SRCOPT

Specifies, only for copying to a source physical to-file, whether new sequence numbers are inserted in the sequence number fields, and whether the date fields are set to zero. If *SEQNBR is specified, the SRCSEQ parameter gives the values that control the new sequence numbers created in the records copied to each to-file member.

***SAME:** The value does not change.

***SEQNBR:** New source sequence numbers are inserted in the records copied to the to-file. The new sequence numbers are controlled by the SRCSEQ parameter value. This value is valid only if the to-file is a source physical file. If *SEQNBR is specified, *DATE can also be specified.

***DATE:** The source date field is set to zero in the records copied to the to-file. This value is valid only if the to-file is a source physical file. If *DATE is specified, *SEQNBR can also be specified.

SRCSEQ

Specifies, only when SRCOPT(*SEQNBR) is also specified, the sequence number that is given to the first record copied to the to-file, and what increment value is used to renumber all other records copied. This parameter allows the copied file to have as many as 999 999 records with unique sequence numbers. If SRCOPT(*SEQNBR) is specified, but SRCSEQ is not specified, SRCSEQ(1.00 1.00) is assumed; the copied records are renumbered sequentially beginning with 1.00, and the whole number increment of 1 is used.

If SRCOPT(*SEQNBR) is specified and the maximum sequence number value of 9999.99 is reached, then all remaining records copied to the to-file member also have a sequence number value of 9999.99.

Element 1: Starting Value

1.00: The first source record copied to the to-file has a sequence number of 0001.00.

starting-value: Specify a value ranging from 0000.01 through 9999.99 that is the sequence number of the first source record copied to the to-file. A whole number of no more than four digits and/or a fraction of no more than 2 digits is specified. If the starting value contains a fraction, a decimal point must be used. Examples are .01 and 3250.4. (If a value has a fraction of .00, such as 5000.00, it is specified without the fraction; either 5000 or 5000.00 is valid.)

Element 2: Increment Value

1.00: The copied source records are renumbered in the to-file with whole number increments of 1. (1.00, 2.00, 3.00, ...)

increment-value: Specify a value ranging from 0000.01 through 9999.99 that is used as the increment value for renumbering all source records copied after the first record. A whole number of no more than four numbers and/or a fraction of no more than two numbers can be specified. If the increment value contains a fraction, a decimal point must be used. For example, if SRCSEQ(5000 10) is specified, the first record copied to the file is numbered 5000.00, the second is 5010.00, the third is 5020.00, and so forth. If SRCSEQ(1 .25) is specified, the copied records are numbered 1.00, 1.25, 1.50, 1.75, 2.00, and so forth.

OUTFMT

Specifies, if TOFILE(*PRINT) is specified, whether the copied records are printed in character or hexadecimal format.

Note: This parameter is used only when TOFILE(*PRINT) is specified or the PRINT parameter specifies *EXCLD, *COPIED, or both. This parameter has no effect on the printed output when the to-file is a program-described printer file. The OUTFMT parameter is ignored when the copy operation does not need to produce a formatted printer file.

***CHAR:** Records are printed in character format only.

***HEX:** Records are printed in both character format and hexadecimal format.

ERRLVL

Specifies the maximum number of recoverable read or write errors for the file that are tolerated during the copy operation for a single database from-file member or tape from-file label identifier. This parameter is ignored if the from-file is not a physical, logical, or tape file and the to-file is not a physical file.

0: If any recoverable error occurs, the copy operation ends at the file member in which the error occurs.

***NOMAX:** No maximum number of errors is specified, and all recoverable errors are tolerated. The copy operation continues regardless of the number of recoverable errors found.

number-of-errors: Specify a value that specifies the maximum number of recoverable errors that is allowed in each from-file member or label that is copied. If one

more error occurs than the value specified here, the copy operation is ends.

COMPRESS

Specifies whether the to-file contains a compressed form of the from-file. Compression occurs when deleted records in the from-file are not copied to the to-file. COMPRESS(*NO) is used to copy both deleted and nondeleted records only when the from-file and to-file are both physical files. If MBROPT(*ADD) is specified with COMPRESS(*YES), deleted records that existed in the to-file member before the copy operation are not compressed.

***YES:** The records copied to the to-file are compressed. Deleted records that exist in the from-file are not copied to the to-file. Only nondeleted records are copied, and they are renumbered consecutively in the to-file. That is, the relative record numbers of all nondeleted records that occur after the first deleted record in the from-file are different in the to-file. No physical record data, such as source sequence numbers, is changed by the copy operation as a result of specifying COMPRESS(*YES). If from-file is delete-capable and the to-file is nondelete-capable, COMPRESS(*YES) must be specified.

***NO:** Both the deleted and nondeleted records are copied to the to-file. If the from-file is a database file that is copied in arrival sequence, the relative record numbers in the from-file are not changed in the to-file if MBROPT(*REPLACE) is also specified. If the from-file is a database file that is copied in keyed sequence (no deleted records are contained in an access path), COMPRESS(*NO) is ignored.

Note: If the from-file is a keyed physical file and a record number value is specified for either the FROMRCD or TORCD parameters, the from-file is copied in arrival sequence; therefore, the COMPRESS parameter determines whether deleted records are copied. If COMPRESS(*NO) is specified, the default value must be used for the INCCHAR and INCREL parameters.

Table 20 (Page 1 of 2). Files Used by CPYF Parameters

	Database Files[6]				Device Files							
	Physical		Logical		Diskette		Tape		Printer[2]		Inline Data	
Parameter	From	To	From	To	From	To	From	To	From	To	From	To
FROMFILE	X		X		X[5]		X				X	
TOFILE		X				X[5]		X		X		
FROMMBR	X		X		X		X					
TOMBR		X				X		X				
MBROPT		X										
CRTFILE	X[1]	X[1]	X[1]									
PRINT	X[2]	X[2]	X[2]		X[2]	X[2]	X[2]	X[2]		X[2]	X[2]	
RCDFMT			X									
FROMRCD	X		X[3]		X		X				X	
TORCD	X		X[3]		X		X				X	
FROMKEY	X		X									
TOKEY	X		X									
NBRRCDS	X		X		X		X				X	
INCCHAR	X		X		X		X				X	
INCREL	X		X									
FMTOPT	X	X	X									
SRCOPT		X										
SRCSEQ		X										
OUTFMT	X[2]	X[2]	X[2]		X[2]	X[2]	X[2]	X[2]		X[2]	X[2]	
ERRLVL	X	X	X				X					

Table 20 (Page 2 of 2). Files Used by CPYF Parameters

	Database Files[6]				Device Files							
	Physical		Logical		Diskette		Tape		Printer[2]		Inline Data	
Parameter	From	To	From	To	From	To	From	To	From	To	From	To
COMPRESS	X[4]	X[4]										

[1] If the to-file does not exist before the copy operation begins, and the from-file is either a physical or logical file, CRTFILE(*YES) can be specified to allow the copy operation to create a physical file for the to-file.

[2] A program-described printer file can be specified for the to-file to create a printout that has no special formatting or page headings, or TOFILE(*PRINT) can be used to create a formatted printout. PRINT(*EXCLD) or PRINT(*COPIED) may be specified to create a formatted printout for any copy operation. When a printout is requested (by either TOFILE(*PRINT) or by the PRINT parameter), the OUTFMT parameter specifies whether the data is printed in character form or both character and hexadecimal form.

[3] The FROMRCD and TORCD parameter values are valid for a from-file that is a logical file if it has only an arrival sequence access path (no keyed access path).

[4] Cannot copy from a delete capable file to a file that is not delete capable unless COMPRESS(*YES) is specified.

[5] If both the from-file and to-file are diskette files, the to-file must be spooled (SPOOL(*YES) is specified on CRTDKTF, CHGDKTF, or OVRDKTF command).

[6] DDM files can be either physical or logical files.

Examples

The following examples of the CPYF command show the type(s) of files that are copied, and the function provided by various parameters.

Example 1: Physical File to Physical File

```
CPYF   FROMFILE(PERSONNEL/PAYROLL)
  TOFILE(TESTPAY/PAYROLL)  MBROPT(*ADD)
  CRTFILE(*YES)  ERRLVL(10)
```

This command copies all of the records in the physical file named PAYROLL in the PERSONNEL library to the file PAYROLL in the TESTPAY library. If the from-file contains more than one member, only the first member is copied. If TESTPAY/PAYROLL does not exist, it is created before the records are copied and a member with the same name as the from-file is added to TESTPAY/PAYROLL to receive the copied records.

Because MBROPT(*ADD) is specified, the copied records are added to any existing records in the to-file member. Because RCDFMT(*NONE) is assumed, the to-file TESTPAY/PAYROLL must have the same record format as the from-file. If the to-file (TESTPAY/PAYROLL) is created by the copy operation, it will have the same record format and access path as the from-file (PERSONNEL/PAYROLL). If more than ten recoverable errors occur during the copy operation, the operation ends.

If FROMMBR(*ALL) and TOMBR(*FROMMBR) had also been specified, all of the members in the from-file would be copied to corresponding members (having the same names) in the to-file. For each from-member that has no corresponding to-member, a member is added to the to-file and all the records in the from-member are copied to the new member. For each to-member that already exists, only new records are added to the member. No updates are made to existing records on any type of copy operation. If the to-file contains members for which there are no corresponding members in the from-file, the to-file contains more members than the from-file after the copy operation.

If more than ten recoverable errors occur within a member being copied, the copy operation ends at that point, and remaining members are not copied. ERRLVL(*NOMAX) can be specified to tolerate all recoverable errors, so the copy operation does not end no matter how many recoverable errors occur in a particular file member.

Example 2: Physical File to Physical File

```
CPYF   FROMFILE(PERSONNEL/EMP1)
  TOFILE(PERSONNEL/VACLEFT)
  FROMMBR(VAC)  MBROPT(*REPLACE)
  FROMKEY(1 X'0008872F')  TOKEY(1 X'0810199F')
  INCREL((*IF VAC *GT 5.0))  FMTOPT(*MAP *DROP)
```

In this example, the to-file (VACLEFT) is an existing physical file, but its record format differs from that of the physical file named EMP1, which is being copied. Both files are in the PERSONNEL library. The from-file contains employee records and has a key (employee number). The records selected in the from-file are those with employee numbers ranging from 008872 through 810199. Only records for employees with more than five days of vacation (VAC) are mapped to the receiving file. Records are selected from member VAC, and they replace existing records in the first member of file VACLEFT.

Because the key for the file is a packed decimal number, the FROMKEY and TOKEY values must be specified as hexadecimal strings, and the leading zeros and hexadecimal sign are required in the value. An alternative way of specifying the same key value range follows:

```
FROMKEY(*BLDKEY 8872)  TOKEY(*BLDKEY 810199)
```

When *BLDKEY is specified, the copy operation converts each number to the format required for the file key definition.

Because only a single value is specified, only one key field is used. The *BLDKEY form of the FROMKEY and TOKEY parameters allows omission of leading zeros and a positive sign value when the key is numeric.

If the key for a file is a composite of more than one key field, the *BLDKEY form is used with a list of values for the FROMKEY and TOKEY parameters. For instance, if the key fields for a file are a sales region (10 characters) and the sales for the last month (7 packed decimal numbers with 2 decimal positions), a complete key is specified in either of the following ways:

```
FROMKEY(*BLDKEY (GEORGIA  99.50))

FROMKEY(2 X'C7C5D6D9C7C9C14040400009950F')
```

When the *BLDKEY form is used, each character field is padded with blanks, and each numeric field is converted to the actual key format with the value shifted left or right to correctly align the decimal point.

Example 3: Physical Data File to Physical Source File

```
CPYF    FROMFILE(MYLIB/DATAFILE)  TOFILE(QIDU/QTXTSRC)
  FROMMBR(A1)  TOMBR(*FROMMBR)
  MBROPT(*REPLACE)  FMTOPT(*CVTSRC)
```

This command copies records from physical file DATAFILE in library MYLIB, which is defined as FILETYPE(*DATA), to physical file QTXTSRC in library QIDU, which is defined as FILETYPE(*SRC). Because the two database files are of different types, FMTOPT(*CVTSRC) must be specified. Records are copied to member A1, which has the same name as the from-file member. Values are assigned to the sequence number source field of the records copied to the source file, starting with 1.00 and incremented by 1.00. If SRCOPT(*SEQNBR) is specified, the SRCSEQ parameter is used to control the sequence numbers that are created. The date source field is always set to zeros.

Example 4: Logical File to Physical File

```
CPYF    FROMFILE(DEPTS/SALES)  TOFILE(DEPTS/YTDSALES)
  FROMMBR(TOTSALES)  TOMBR(MARCH)  RCDFMT(AA)
  NBRRCDS(5)  MBROPT(*REPLACE)
```

This command copies five records from member TOTSALES of logical file SALES (in library DEPTS) to member MARCH in the physical file YTDSALES (in library DEPTS). If member MARCH does not exist, it is created and added to the to-file automatically by the copy operation. Only records from the logical file SALES in library DEPTS that use record format AA are copied, and they are copied to YTDSALES, which has the same format. After the copy operation, the MARCH member contains only five nondeleted records, because all records in that member are first cleared, then only the data in the first five records (in keyed sequence) in the TOTSALES member are copied to it.

Example 5: Device File to a Physical File

```
CPYF    FROMFILE(QDKT)  TOFILE(QGPL/QCLSRC)
  FROMMBR(PAY*)  TOMBR(*FROMMBR)
  MBROPT(*REPLACE)  SRCOPT(*SEQNBR)
  SRCSEQ(1  .25)
```

This command copies records from the generic set of diskette labels with names that start with the characters PAY. They are copied to like-named members in source file QCLSRC in the QGPL library. Even though the to-file is a source file, a diskette file (QDKT) defined as FILETYPE(*DATA) is used as the from-file, because QDKT is more efficient than a device file defined as FILETYPE(*SRC). For each label copied, the sequence number of the first record is 1.00 and is incremented by .25 for each subsequent record. The source date field is automatically set to zeros.

Example 6: Physical File to the Printer

```
CPYF    FROMFILE(TEMPFILE)  TOFILE(*PRINT)
  FROMMBR(EMP1)  FROMKEY(1 448762)
  NBRRCDS(20)  OUTFMT(*HEX)
```

This command copies records from member EMP1 in the file named TEMPFILE. The records are employee records. One key field, the employee number, is used to search the record keys. Twenty records, starting with employee number 448762, are copied to the IBM-supplied printer file QSYSPRT and listed in both character and hexadecimal format. The IBM-supplied printer file is indicated by coding TOFILE(*PRINT).

Example 7: Physical File to a Device File

```
CPYF    FROMFILE(PERSONNEL/PAYROLL)
  TOFILE(DISK1)  FROMMBR(VAC1)
  INCCHAR(NAME 1 *CT SMITH)
  INCREL((*IF VAC *GT 10.5)(*AND HOLIDAYS *EQ 0))
```

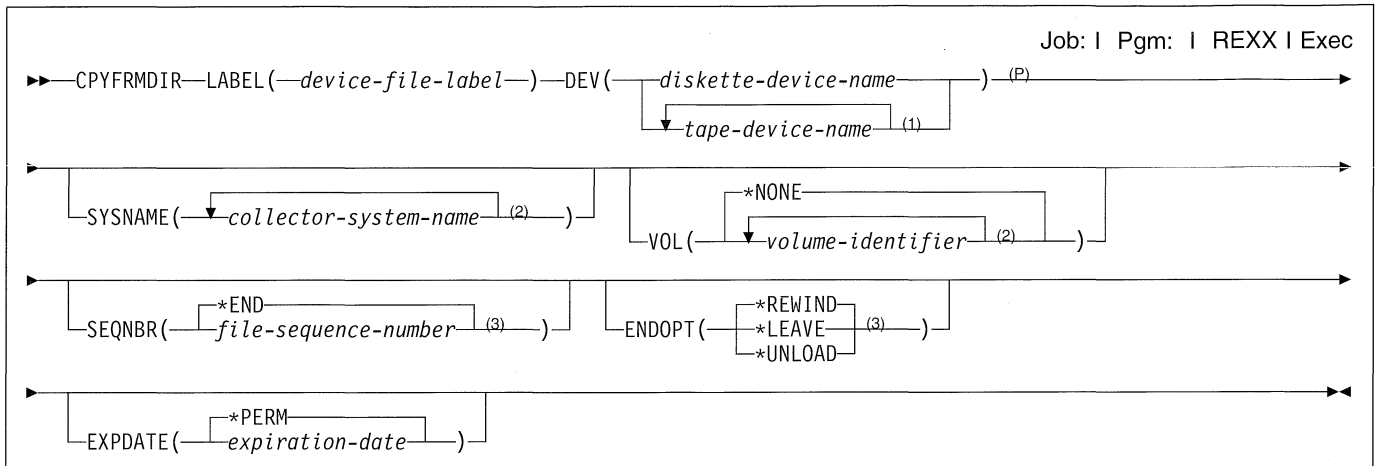
This command copies all employee records of employees whose last name is SMITH and that have accumulated more than ten and a half vacation days, none of which is holidays, from the PAYROLL file in the PERSONNEL library to a diskette. The file member name copied is VAC1. The vacation (VAC) and holiday (HOLIDAYS) fields are defined as packed decimal, but a value is specified in character form on the INCREL parameter. The diskette device file used is DISK1, which contains the label of the file being copied to, and other diskette attributes such as location and volume ID.

Example 8: Physical File to Device Files

```
CPYF    FROMFILE(PERSONNEL/PAYROLL)  TOFILE(DISK1)
  FROMMBR(*ALL)  TOMBR(*FROMMBR)
```

This command copies all members of file PAYROLL in the PERSONNEL library to data files on diskette (device file DISK1). Each from-file member name must be a valid diskette label identifier; if not, use the RNMM (Rename Member) command to rename the members in the from-file before they are copied.

CPYFRMDIR (Copy From Directory) Command



Notes:

1 A maximum of 4 repetitions.

2 A maximum of 50 repetitions.

3 Applies to tape devices only.

P All parameters preceding this point can be specified in positional form.

Purpose

The Copy From Directory (CPYFRMDIR) command is used to copy directory data from the local system to a tape or diskette device. This directory data can then be copied to other remote systems by using the Copy To Directory (CPYTODIR) command on the remote systems. This function allows the remote system to begin a directory shadowing environment with the local system by shadowing changes made to the directory data from the local system.

Caution: Do not use this command as a backup utility to save and restore directory data for data recovery purposes. Follow the normal backup and recovery procedure guidelines described in the *Advanced Backup and Recovery Guide*.

Restriction: The user must have security administrator (*SECADM) authority to use this command.

Required Parameters

LABEL

Specifies the name that identifies the device file label on the tape or diskette to be copied. A maximum of 17 characters can be specified for tape devices and 8 characters for diskette devices.

DEV

Specifies the names of the tape or diskette devices used for the copy operation. Each tape or diskette device name must already be known on the system by a device description.

diskette-device-name: Specify the name of the diskette device to be used for the copy operation.

tape-device-name: Specify the names of one or more tape devices used for the copy operation. If more than one tape device is used, specify the names of the devices in the order in which they are used. Using more than one tape device permits one tape volume to be rewound and unloaded while another tape device processes the next tape volume.

Optional Parameters

SYSNAME

Specifies the names of the remote systems that copies the directory data from the tapes or diskettes created by this command. The names of the remote systems on this parameter are added to the list of system names that are collecting changes to directory data from the local system.

Note: You must include the names of all the remote systems that use the tapes or diskettes created by this command to ensure that all changes to directory data are sent to the remote systems during a normal shadowing session.

VOL

Specifies one or more volume identifiers used by the file. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***NONE:** No volume identifiers are specified for the file. No volume identifiers are checked.

volume-identifier: Specify the identifiers of one or more volumes in the order in which they are placed in a device.

SEQNBR

Specifies the sequence number of the data file on the tape being processed. The four-position file sequence number is read from the first header label of the data file.

***END:** The copy operation begins after the last sequence number on the tape volume.

file-sequence-number: Specify the sequence number of the file that is used. Valid values range from 0001 through 9999.

ENDOPT

Specifies the operation that is automatically performed on the tape volume after the operation ends. If more than one volume is included, this parameter applies only to the last tape volume used; all other tape volumes are rewound and unloaded when the end of the tape is reached.

***REWIND:** The tape is automatically rewound, but not unloaded, after the operation has ended.

***LEAVE:** The tape does not rewind or unload after the operation ends. It remains at the current position on the tape drive.

***UNLOAD:** The tape is automatically rewound and unloaded after the operation ends.

EXPDATE

Specifies the expiration date. The files cannot be over-written until the expiration date. The expiration date must be later than or equal to the current date.

***PERM:** The data file is permanently protected. An expiration date of 999999 is assigned.

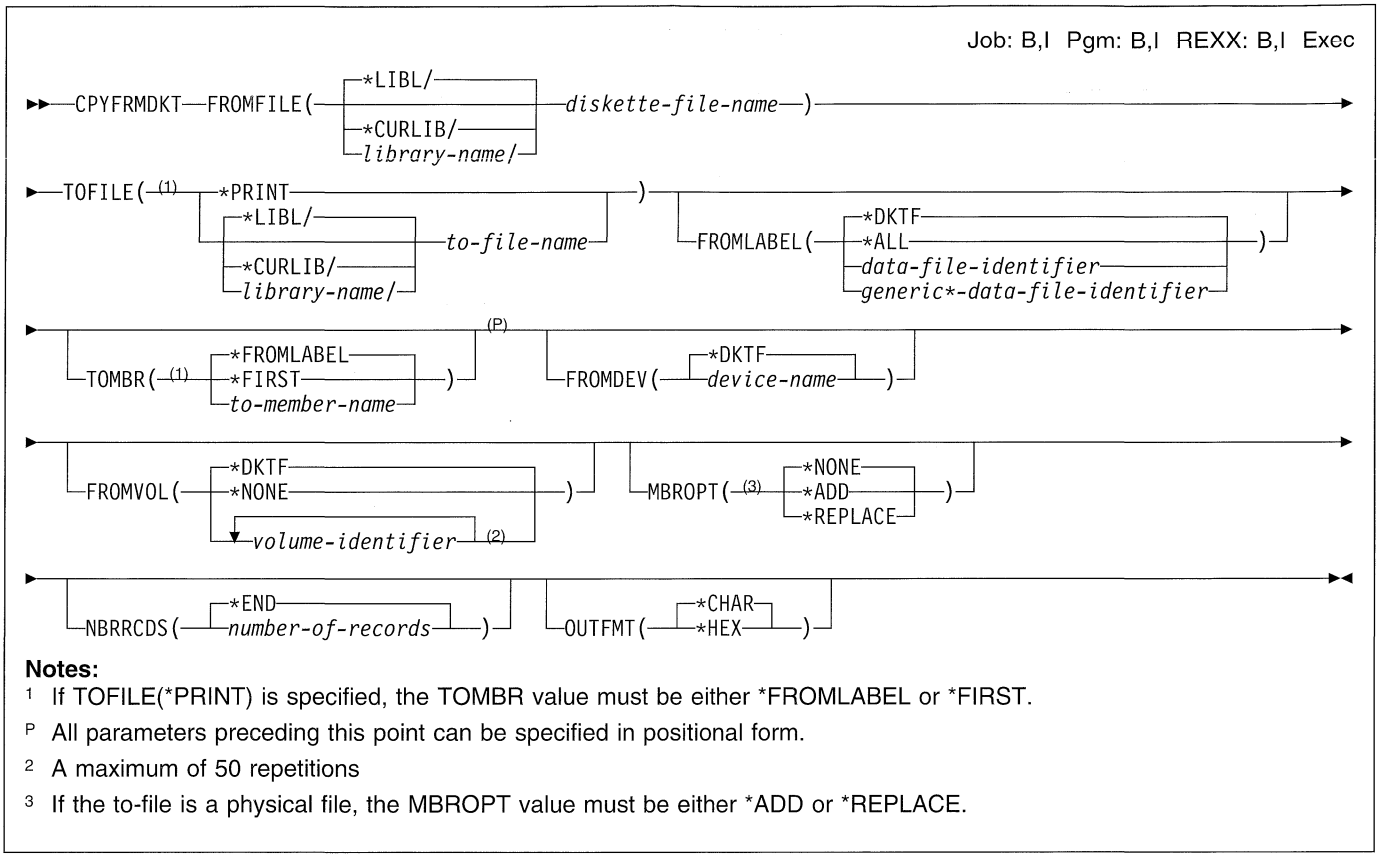
expiration-date: Specify the date when protection for the file ends.

Example

```
CPYFRMDIR   DEV(TAP01)  SYSNAME(CHICAGO NEWYORK)
```

This command copies all of the directory data from the local system to tape device TAP01. CHICAGO and NEWYORK are added to the list of systems that collect changes to the directory data from the local system.

CPYFRMDKT (Copy from Diskette) Command



Notes:

1 If TOFILE(*PRINT) is specified, the TOMBR value must be either *FROMLABEL or *FIRST.

P All parameters preceding this point can be specified in positional form.

2 A maximum of 50 repetitions

3 If the to-file is a physical file, the MBROPT value must be either *ADD or *REPLACE.

Purpose

The Copy from Diskette (CPYFRMDKT) command copies one or more data files from diskette to an output file or to the printer. The from-file must be a diskette file for this command, but the to-file can be a physical, DDM, program-described printer, tape, or diskette file, or *PRINT to print the records using the IBM-supplied printer file QSYSPRT.

Note: For more information on DDM files, see the *DDM Guide*.

This command offers a subset of the parameters available on the Copy File (CPYF) command. If you need parameters that are not available on the CPYFRMDKT command, you can either use overrides for the from-file or to-file, or use the CPYF command or a combination of file overrides with the Copy File (CPYF) command.

One label, a generic set of labels, or all labels from the diskette are copied. The *Data Management Guide* has a complete description of the combinations allowed and how to specify them.

The to-file must exist when the CPYFRMDKT command is started. This command does not create the to-file, but it *does* add a member to an existing physical file if the member does not already exist in the to-file.

Note: This command cannot be used to copy save/restore type files.

Restriction: A file's open data path (ODP) cannot be shared with any other program in the job (routing step) during the copy operation.

Required Parameters

FROMFILE

Specifies the qualified name of the diskette device file that contains the copied records.

The name of the diskette file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

diskette-file-name: Specify the name of the diskette device file that contains the copied records.

TOFILE

Specifies the qualified name of the file that receives the copied records.

Note: If no library qualifier is given, *LIBL is used to find the file. The device file can be a diskette, tape, or program-described printer file. If a diskette file is used for the TOFILE parameter, the diskette spool writer must not be active and the diskette file must be defined with SPOOL(*YES), because the system is not able to allocate the same diskette device for both the from-file and the to-file to do the copy operation.

***PRINT:** The records are copied to the IBM-supplied printer file QSYSPRT, and the file is formatted according to the OUTFMT parameter. The IBM-supplied printer file QSYSPRT may not be overridden to a different file name, and it must have the RPLUNPRT(*YES) and CTLCHAR(*NONE) attributes.

The name of the to-file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

to-file-name: Specify the name of the physical file or device file that receives the copied records. If no library qualifier is given, *LIBL is used to find the file.

Optional Parameters

FROMLABEL

Specifies the label identifier of a single-diskette data file or the generic identifier for a group of diskette data files that are copied, or indicates that all data files are copied from the diskette.

***DKTF:** Specifies that the data file label identifier in the diskette device file description is used to identify the file on the diskette that is copied (it can also be specified in an override for the from-file).

***ALL:** All data files on the diskette volumes are copied. If the to-file is either a spooled diskette or database physical file, the records can be copied to corresponding diskette labels or physical file members in the to-file with the same name (by specifying TOMBR(*FROMLABEL)), or it can be copied to a single label or member that contains a concatenation of all records from all data files copied from a diskette. If the to-file is a printer file, each data file is copied to a separate spooled file. If TOFILE(*PRINT) is specified, all the data files on the diskette are copied to a single-print output file and the records for each data file that is copied begins on a new print page.

If FROMLABEL(*ALL) is specified and a LABEL parameter value is also specified on an Override Diskette File (OVRDKTF) command, only the single-file label identifier specified in the override is copied.

data-file-identifier: Specify the label identifier of the data file that is read from the diskette. If a different LABEL parameter value has been specified on an OVRDKTF command, the label identifier specified on the OVRDKTF command is used instead of the value specified on this parameter.

generic-data-file-identifier:* Specify the generic label identifier of the data files that are copied from the diskette. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be copied only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

If a generic identifier is specified for the FROMLABEL parameter and a LABEL parameter value is also specified on an OVRDKTF command, only the single-file label identifier specified on the override is copied.

If one of the files being copied from a diskette is continued onto another volume and FROMLABEL(*ALL) or a generic label identifier is specified, all the files on the continuation volume are processed. The system attempts to copy files from all diskettes until it completes processing for a diskette with no copied file that continues onto another volume.

TOMBR

Specifies the name of the file member that receives the copied records.

Note: If the TOFILE is a diskette or tape device file, TOMBR specifies the label identifier of the data file to which the records are copied. If the TOFILE is a printer file or *PRINT, then *FROMLABEL or *FIRST must be specified. A physical file member is added with the name specified by this parameter (including a name implied by *FROMLABEL) if one does not exist.

***FROMLABEL:** Specifies that all files which are identified by the data file identifier specified in the FROMFILE parameter, are copied to corresponding members or diskette or tape labels in the physical to-file. If a member or file with a corresponding name does not exist in the to-file, then a member or a file with that name is added to a physical to-file.

If a single data file identifier was specified as a value for the FROMLABEL parameter, then a member in the

to-file with the same name receives the copied records. If a generic data file identifier or *ALL was specified as a value for the FROMLABEL parameter, each file label in the from-file is copied to a corresponding member or label in the to-file. If the to-file is a tape file and *FROMLABEL is specified, then a single-data file identifier or *DKTF must be specified for the FROMLABEL parameter. If the to-file is a tape or diskette device file, the label in the device file description is used.

***FIRST:** The first member in the physical file receives the copied records.

to-member-name: Specify the name of the physical file member or the file label identifier of the diskette or tape data file that receives the copied records. If the label identifier for the tape file is more than 10 characters long or contains special characters, then it must be specified on the Create Tape File (CRTTAPF), Change Tape File (CHGTAPF), or Override Tape File (OVRTAPF) command.

FROMDEV

Specifies the name of diskette devices from which the diskette from-file is copied.

***DKTF:** The value specified in the diskette device file is used to indicate the devices used.

device-name: Specify the names of diskette devices used when copying records from the from-file. The order in which the device names are specified is the order in which the tapes on the devices are read. Each device name must already be known on the system by a device description.

FROMVOL

Specifies the diskette that is used.

***DKTF:** The diskette volume identifiers in the diskette device file are used to identify the diskette file that is copied (it can also be specified in an override for the from-file).

***NONE:** No volume identifier checking is done.

volume-identifier: Specify up to 50 volume identifiers used to identify the diskettes that are copied. Each identifier can have 6 alphanumeric characters or less.

MBROPT

Specifies whether the new records replace or are added to the existing records.

Note: If the to-file is a device file, this parameter is ignored. If the to-file is a physical file, this parameter is required.

***NONE:** No records are added or replaced in a member. This value is valid only for a copy to a device file.

***ADD:** The system adds the new records to the end of the existing records.

***REPLACE:** The system clears the existing member and adds the new records.

NBRRCDS

Specifies the number of records copied to the to-file.

***END:** Records are copied until the end-of-file condition is indicated.

number-of-records: Specify the number of records, ranging from 1 to 2147483647, that are copied to the to-file. Fewer records are copied if an end-of-file condition occurs before the specified number of records have been copied.

OUTFMT

Specifies, if TOFILE(*PRINT) is specified, whether the copied records are printed in character or hexadecimal format.

***CHAR:** Records are printed in character format only.

***HEX:** Records are printed in both character format and hexadecimal format.

Examples

Example 1: Copying Records to a Database File

```
CPYFRMDKT  FROMFILE(QDKT)  TOFILE(MASTER/PAYROLL)
  FROMLABEL(MONTH1)  MBROPT(*REPLACE)
```

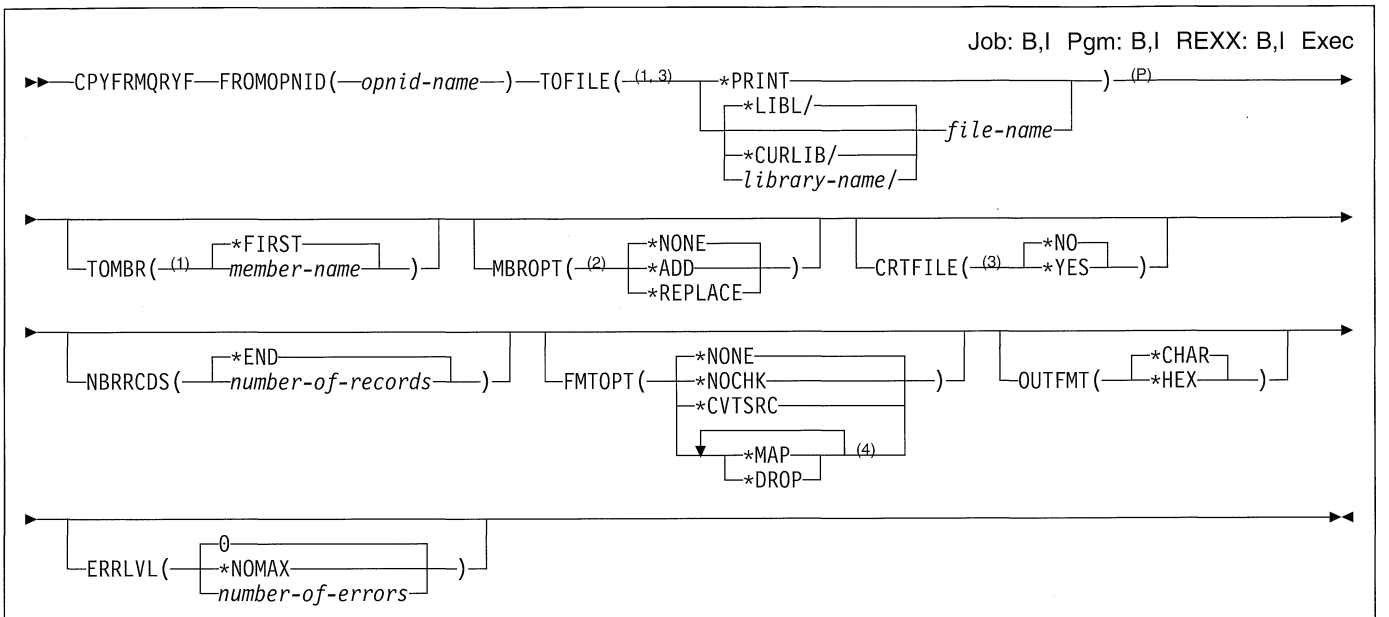
This command copies records from a diskette using the diskette device file QDKT. The diskette device specified on the QDKT file description is created. The data file on the diskette that is copied is identified by label MONTH1. The records are copied to the physical database file PAYROLL in library MASTER and replaces the existing records in member MONTH1 (which is implied by the parameter default of TOMBR(*FROMLABEL)).

Example 2: Printing Copied Records

```
CPYFRMDKT  FROMFILE(QDKT)  TOFILE(*PRINT)
  FROMDEV(DKT2) FROMLABEL(MONTH*)  FROMVOL(PAY1)
```

This command copies from a diskette, by using diskette device file QDKT and the diskette device DKT2, the generic set of labels that start with the characters MONTH. The diskette volume identifier is specified on the command, which eliminates the need for a separate override command. The records are listed on the printer by using IBM-supplied printer file QSYSPRT and printed in character format, which is the default for the OUTFMT parameter.

CPYFRMQRYF (Copy From Query File) Command



Notes:

1 If TOFILE(*PRINT) is specified, the TOMBR value must be *FIRST.

2 If CRTFILE(*YES) is specified and the to-file does not exist, MBROPT(*ADD) is assumed. If the to-file is an existing physical file, the MBROPT value must be either *ADD or *REPLACE.

3 If CRTFILE(*YES) is specified and the to-file does not exist, a library name (not *LIBL) must be specified for the TOFILE parameter.

P All parameters preceding this point can be specified in positional form.

4 A maximum of 2 repetitions

Purpose

The Copy from Query File command copies a set of database records that satisfies a database query request to a physical file, a program-described printer file, an IBM-supplied printer file (QSYSPRT which is used when *PRINT is specified), a DDM file, a tape file, or a diskette file. The records are copied from the open query file associated with an open identifier. The open identifier must name an open query file that was opened for input, update, or all operations.

Note: For more information on DDM files, see the *DDM Guide*.

This command can:

- Add records to an existing physical file member or replace contents of a receiving physical file member (MBROPT parameter).
- Copy records from an open query file format that is different than the to-file record format, and convert records when copying to a source physical file. When the formats are different, the copy operation can:
  - Map fields that have the same name in the open query file format and the to-file record format
  - Drop fields from the open query file format that do not exist in the to-file record format
  - Copy the records directly, disregarding the differences between the open query file format and the to-file record format
- Select a printout format when TOFILE(*PRINT) is specified. The records can be listed in character format, or in both character and hexadecimal format (OUTFMT parameter).
- If the to-file does not exist before the copy operation, create the file (CRTFILE parameter) so that it has the same format as the open query file. Only the name, type, length, and decimal positions of each field in the open query format are used.
- Copy a specified number or all of the records from an open query file depending if NBRRCDS(*END) was used or if a specific number was specified on the NBRRCDS parameter.

Additional information about CPYFRMQRYF is in the *Data Management Guide* and the *Database Guide*.

Restrictions:

1. The open query file used by the CPYFRMQRYF command must not use any DDM files (specified on the FILE parameter of the OPNQRYF command).
2. A member cannot be copied to itself. This restriction means that a member specified by the TOFILE and TOMBR parameters of the CPYFRMQRYF command cannot have the same name as any member specified on the FILE parameter of the OPNQRYF command, nor can the member have the same qualified name as any physical file members used by logical files that are on the FILE parameter of the OPNQRYF command.

Required Parameters

FROMOPNID

Specifies the name used on the OPNQRYF command for identifying the open identifier for the query file. The open identifier (specified by the OPNID parameter of OPNQRYF) names an open query file that allows input, update, or all operations as specified by the OPNID and OPTION parameters on the OPNQRYF command.

TOFILE

Specifies the qualified name of the file that receives the copied records.

Note: A device file can be a diskette file, tape file, or a program-described printer file.

***PRINT:** The data is copied to the IBM-supplied printer device file, QSYSPRT, and the file is formatted according to the OUTFMT parameter.

The IBM-supplied printer file, QSYSPRT, may not be overridden to a different file name, and it must have the RPLUNPRT(*YES) and CTLCHAR(*NONE) attributes.

The name of the file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

file-name: Specify the qualified name of the physical file or device file that receives the copied records. If no library qualifier is specified, *LIBL is used to locate the file. However, if CRTFILE(*YES) is specified and the specified file cannot be found, the file name must be qualified with a library name. When the physical to-file is created, it is placed in the specified library.

Optional Parameters

TOMBR

Specifies the name of the file member that receives the copied records.

Note: If the to-file is a printer file, the value of the TOMBR parameter must be *FIRST.

When a physical file is created by the copy operation for the to-file and TOMBR(*FIRST) is specified, the to-file file name is used as the file member name in the created file.

***FIRST:** The first member in the database file receives the copied records.

member-name: Specify the name of the physical to-file member, or the label identifier of the diskette to-file or tape to-file that receives the records. If a member with the specified name does not already exist in the physical to-file, the copy operation attempts to add a member with the specified name to the file.

MBROPT

Specifies whether the new records replace or are added to the existing records.

Note: If the copy is to an existing physical file, this parameter must specify either *ADD or *REPLACE. If the to-file does not exist but CRTFILE(*YES) is specified, the copy operation assumes MBROPT(*ADD) for all records copied to the file after it is created, regardless of the value specified on this parameter. The copied records are always physically added to the end of a database file member in the same order that they are retrieved from the open query file.

***NONE:** The MBROPT parameter does not apply to this copy operation. When the to-file is an existing physical file, MBROPT(*NONE) is not allowed; either *ADD or *REPLACE must be specified to indicate whether records should be added or replaced in each to-file member used.

***ADD:** The system adds the new records to the end of the existing records.

***REPLACE:** The system clears the existing member and adds the new records.

CRTFILE

Specifies whether a physical file is created to receive the data if the to-file does not exist. If the to-file already exists when this command is started, this parameter is ignored.

***NO:** The to-file must exist when this command is started. A physical file is not created to receive the data.

***YES:** If the specified to-file does not exist, a physical file is created that has the name specified on the TOFILE parameter. In addition to the normal copy operation validity checks, the following special conditions must all be true for the copy operation to create a to-file:

- A library name must be specified on the TOFILE parameter (the default value, *LIBL, is not allowed).

- There cannot be an override to a different file or library name; the values specified on this command for the to-file must be used.
- The user running the CPYFRMQRYF command must be authorized to add the file to the TOFILE library and must have operational authority to the CRTPF command.

NBRRCDS

Specifies the number of records copied to the to-file.

Note: The first record copied is the record at the start of the open query file access path.

***END:** Records are copied until the end-of-file condition is indicated.

number-of-records: Specify the number of records, ranging from 1 to 2147483647, that are copied to the to-file. Fewer records are copied if an end-of-file condition occurs before the specified number of records have been copied.

FMTOPT

Specifies, when the open query file is copied to a physical to-file, what field-level record format processing (if any) is done. If the to-file is a source physical file, *CVTSRC must be specified on this parameter. The CCSIDs for character and DBCS fields in the open query format are determined by the CCSID of the job in which the Open Query File (OPNQRYF) command is run. All CCSIDs other than 65535 are reset to the job CCSID of the OPNQRYF command, unless the OPNQRYF job CCSID is 65535, in which case the CCSIDs are unchanged. If the open query file format and to-file record formats are identical and the to-file is a data physical file, any FMTOPT value except *CVTSRC can be specified to perform the copy operation.

Note: Change the job CCSID to 65535 before running the OPNQRYF command if you plan to use the CPYFRMQRYF command.

***NONE:** No field mapping or field dropping is done during the copy operation. This value is valid only if the open query file and to-file have the same record format, or if the to-file is not a database file. The record formats are the same only if every field exists in both the open query file and to-file formats, and has the same starting position and attributes in both formats. Attributes include CCSIDs, whether or not a field is null-capable, and the date/time format and separator (if the field is a date/time field). Null values are copied if *NONE is valid and both files are database files.

***NOCHK:** If the record formats of the open query file and the to-file are different, the copy operation continues despite the differences. Record data is copied directly (left to right) from one file to the other. If this value is specified, null values are ignored, no conversion of date/time data occurs, and no CCSID conversions are done.

***CVTSRC:** This value is used to copy to a source file. It is valid only when the to-file is a source file. If the to-file is a source file, sequence number and date fields are added, and the open query file record data is copied to the source data part of each to-file record. Null values are ignored and no conversion of date/time data occurs.

***MAP:** Fields with the same name in the open query file and to-file record formats are copied, and fields in the to-file that do not exist in the open query file format are set to the default value specified on the DFT keyword for the data description specification (DDS) of the to-file (or zero for numeric fields, blanks for character fields, current date/time for date/time fields, or the null value for null-capable fields). If *MAP is specified, *DROP can also be specified. Mapped fields may have different starting positions in the open query file and to-file record formats. *MAP allows for CCSID conversions, the conversion of date/time data and for the copying of null values.

***DROP:** This value must be specified for field-level mapping if any of the field names in the open query file record format do not exist in the to-file format. If *DROP is specified, *MAP can also be specified. When *DROP is specified, all the field names that exist in both record formats must have the same attributes and relative positions in the open query file and to-file record formats, or *MAP must also be specified. Null values are copied if *DROP is valid.

OUTFMT

Specifies, if TOFILE(*PRINT) is specified, whether the copied records are printed in character or hexadecimal format.

***CHAR:** Records are printed in character format only.

***HEX:** Records are printed in both character format and hexadecimal format.

ERRLVL

Specifies the maximum number of recoverable read or write errors that are tolerated for the file during the copy operation. The recoverable error count is reset at the beginning of each CPYFRMQRYF operation. If the number of recoverable errors handled is larger than the number specified on the ERRLVL parameter, the copy operation ends and a message is sent.

0: If a recoverable error occurs, the copy operation ends.

***NOMAX:** No maximum number of errors is specified. All recoverable errors are tolerated. The copy operation continues regardless of the number of recoverable errors found.

number-of-errors: Specify a value that specifies the maximum number of recoverable errors that is allowed for the copy operation. If one more error occurs than the value specified here, the copy operation ends.

Examples

Example 1: Building a File with a Subset of Records

```
OPNQRYF  FILE(CUSTOMER/ADDRESS)
  QRYSLT('STATE *EQ "TEXAS"')
CPYFRMQRYF  FROMOPNID(ADDRESS)
  TOFILE(TEXAS/ADDRESS)  CRTFILE(*YES)
```

These commands create a file from the CUSTOMER/ADDRESS file containing records that have a value of Texas in the STATE field.

Example 2: Printing Records Based on Selection

```
OPNQRYF  FILE(FILEA)  QRYSLT('CITY *EQ "CHICAGO"')
CPYFRMQRYF  FROMOPNID(FILEA)  TOFILE(*PRINT)
```

These commands print all records from FILEA where the value of the CITY field is Chicago.

Example 3: Copying a Subset of Records to a Diskette

```
OPNQRYF  FILE(FILEB)  QRYSLT('FIELDB *EQ "10"')
OPNID(MYID)  CPYFRMQRYF  FROMOPNID(MYID)
TOFILE(DISK1)
```

These commands copy to a diskette file all records from FILEB where the value of FIELDB is 10.

Example 4: Creating a Copy of the Output from a Dynamic Join

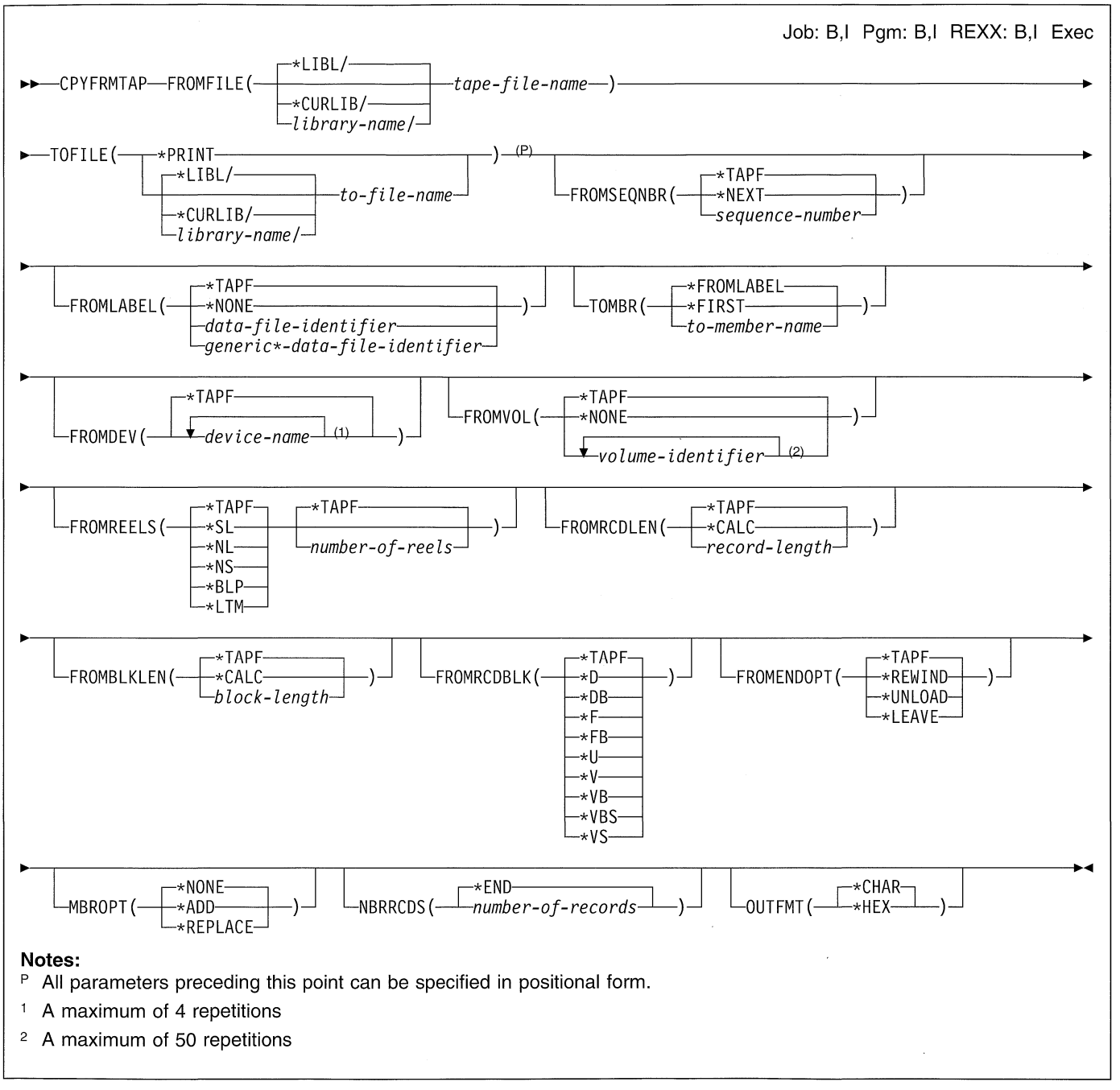
Assume a user wants to create a physical file with the format and data of FILEA and FILEB. Assume the files contain the following fields:

```
FILEA       FILEB      JOINAB
-----       -----      ------
Cust        Cust       Cust
Name        Amt        Name
addr                   Amt
```

```
OPNQRYF  FILE(FILEA FILEB)  FORMAT(JOINAB)
  JFLD((FILEA/CUST FILEB/CUST))
  MAPFLD((CUST 'FILEA/CUST'))  OPNID(QRYFILE)
CPYFRMQRYF  FROMOPNID(QRYFILE)
  TOFILE(MYLIB/FILEC)  CRTFILE(*YES)
```

These commands join FILEA and FILEB and save a copy of the results in a new physical file MYLIB/FILEC. The format of the file will be like JOINAB. The file will contain the data from the join of FILEA and FILEB using the Cust field. File FILEC in library MYLIB can be processed like any other physical file with CL commands (for example, Display Physical File Member (DSPPFM)) and utilities (for example, Query/400).

CPYFRMTAP (Copy from Tape) Command



Notes:

| P All parameters preceding this point can be specified in positional form.

1 A maximum of 4 repetitions

2 A maximum of 50 repetitions

Purpose

The Copy from Tape (CPYFRMTAP) command copies records from a tape file to an output file or to a printer. The from-file must be a tape file, but the to-file can be a physical file, a DDM file, diskette file, tape file, or program-described printer file. To obtain a formatted printer file of the records using the IBM-supplied printer file QSYSPRT, TOFILE(*PRINT) is specified.

Note: For more information on DDM files, see the *DDM Guide*.

This command offers a subset of the parameters available on the Copy File (CPYF) command, along with more specific tape-oriented parameters. If parameters are needed that are not available on the CPYFRMTAP command, the user can either use overrides for the from-file or to-file, or use the CPYF command or a combination of file overrides with the CPYF command. Only a single tape file (sequence number) can be copied with one call of this command.

The to-file must exist when the CPYFRMTAP command is started. This command does not create the to-file, but it

does add a member to an existing physical file if the member does not already exist in the to-file.

Restrictions:

1. A file's open data path (ODP) cannot be shared with any other program in the job (routing step) during the copy operation.
2. Non-labeled tapes cannot be duplicated to 1/4 inch or 8mm cartridge devices.

Required Parameters

FROMFILE

Specifies the qualified name of the tape device file that contains the records that are copied.

The name of the tape file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

tape-file-name: Specify the name of the tape device file containing records to be copied.

TOFILE

Specifies the qualified name of the file that receives the copied records.

Note: If no library qualifier is given, *LIBL is used to find the file. The device file can be a diskette, tape, or program-described printer file.

***PRINT:** The records are copied to the IBM-supplied printer file QSYSPRT, and the file is formatted according to the OUTFMT parameter. The IBM-supplied printer file, QSYSPRT, may not be overridden to a different file name, and the RPLUNPRT(*YES) and CTLCHAR(*NONE) attributes must be specified.

The name of the to-file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

to-file-name: Specify the name of the physical file or device file that receives the copied records.

Optional Parameters

FROMSEQNBR

Specifies the sequence number of the tape file that is copied. If the tapes for the from-file have standard labels, the file sequence number is read from the first header label of the data file. When bypass label processing has been specified for the from-file (that is, FROMREELS(*BLP) has been specified on this command or on the command used to describe the tape file) or when tapes with standard labels are not being used, the system uses the tape marks and the value specified on this parameter to locate the correct data file that is copied.

***TAPF:** The value specified in the tape device file is used.

***NEXT:** The next file in the sequence is processed. If the tape is currently in a position that is prior to the first file, the first file on the tape is processed.

sequence-number: Specify the sequence number of the file on the tape that is copied. For a labeled tape file, the label specified in the FROMLABEL parameter must be found at this sequence number. Valid sequence numbers range from 1 through 9999.

FROMLABEL

Specifies the data file identifier of a file that is copied from the tape. The file identifier specified must be found at the sequence number specified in the FROMSEQNBR parameter.

***TAPF:** The value specified in the tape device file or override is used.

***NONE:** The file identifier is not specified. No check is made for the file identifier in the labels file specified by the FROMSEQNBR parameter. Any label identifier is accepted.

data-file-identifier: Specify the name of the file label that is copied from the tape from-file. A label may be up to 17 characters in length.

generic-data-file-identifier:* Specify the generic name of the data file identifier. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be copied only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

TOMBR

Specifies the name of the file member that receives the copied records.

Note: Either *FROMLABEL or *FIRST must be specified for this parameter if the to-file is a printer device file or if *PRINT is specified for the TOFILE parameter. If a member (specified by name or implied by a parameter value of *FROMLABEL) does not exist in a physical to-file, it is added to the to-file by the copy operation.

***FROMLABEL:** The file specified by the FROMLABEL and FROMSEQNBR parameters is copied into a corresponding member or label in the to-file.

The name of the from-file tape label identifier is used as the member or label identifier for a physical to-file, diskette to-file, or tape to-file. If the to-file is a diskette or tape file, the from-file label identifier is used without modification.

If the to-file is a database file, the last 10 characters that appear before all consecutive blanks for the from-file label are used for the to-file member name. If the last 10 characters that appear before all consecutive blanks are not valid, then the characters to the right of the last period (.) are used for the to-file member name.

If the from-file is a nonlabeled tape file, then a to-file member or label name is created that corresponds to the data file on the tape from-file in the form of CPYnnnn, where nnnn is the tape sequence number of the data-file. If the to-file is a tape or diskette device file, the label in the device file description or override is used.

***FIRST:** The first member in the physical to-file receives the copied records.

to-member-name: Specify the name of the physical file member or the label of a diskette or tape file that receives the records copied from the from-file. If the tape label identifier is longer than 10 characters or contains special characters, it must be specified on a CRTTAPF, CHGTAPF, or OVRTAPF command before starting the CPYFRMTAP command.

FROMDEV

Specifies the name of up to four tape devices from which the tape from-file is copied.

***TAPF:** The value specified in the tape device file is used.

device-name: Specify the names of diskette devices used when copying records from the from-file. The order in which the device names are specified is the order in which the tapes on the devices are read. Each device name must already be known on the system by a device description.

FROMVOL

Specifies one or more volume identifiers of the tapes used when copying records from the (tape) from-file. These volumes must be placed in the devices in the same order as the identifiers are specified (and in the same order as the device names are specified in the FROMDEV parameter).

***TAPF:** The value specified in the tape device file or override is used.

***NONE:** No tape volume identifiers are specified for this file. No volume checking is done beyond verifying that the correct label type volume is placed in the device. The maximum number of reels processed for a *NL, *NS, or *BLP from-file when FROMVOL(*NONE) is specified is determined by the number-of-reels element of the FROMREELS parameter.

volume-identifier: Specify no more than 50 volume identifiers from which the tape file is copied. The identifiers of volumes must be entered in the same order in which they are placed in the devices. Each identifier can have up to 6 alphameric characters. The maximum number of reels processed for an *NL, *NS, or *BLP input file is determined by the number of volume identifiers in the list. For *SL and *BLP, the system compares the identifiers of the volumes against the identifiers in the list, and sends the operator an inquiry message if the wrong volume is placed in the device.

FROMREELS

Specifies the type of labeling used on the tape reels that contain the from-file. The number of reels value specified on this parameter is not a limiting value for tape with standard labels. For standard-label tape, the labels on the tape indicate an end-of-file condition that limits the number of volumes processed. The number of reels information is used only if there is no list of volume identifiers specified and if the from-file is either *NL, *NS or *BLP. When the number of reels value is used, the volume identifiers on the volumes that are placed in the device are ignored if the from-file resides on labeled tapes. In that case, the order in which the reels are placed in the device must be checked by the operator.

Element 1: Label Type

***TAPF:** The value specified in the tape device file or override is used.

***SL:** Each tape volume from which records are copied has standard labels.

***NL:** The tape volumes, from which records are copied, have no labels. Tape marks are used to indicate the end of each data file on the tape.

***NS:** Each tape volume from which records are copied has nonstandard labels. Any file label information contained on the tape is ignored; instead, tape marks are used to determine the start and end of a data file. Only a single data file can exist on a nonstandard tape reel (volume).

***BLP:** Standard label processing is bypassed when copying the from-file volumes. Although each reel is checked for a standard label, all other volume label information and file labels on the tape are ignored. Only the tape marks are used when the tape volumes are being copied. Bypass label processing can be used if the names of the volumes are incorrect, or if some of the information in the file labels is incorrect.

***LTM:** The volumes of the from-file have no labels but have a single leading tape mark before the first file on the tape.

Element 2: Number of Reels

***TAPF:** The value specified in the tape device file or override is used.

number-of-reels: Specify the maximum number of reels used when copying records from the from-file for a *NL, *NS, *BLP, or *LTM tape file when there is no list of volume identifiers specified. If the next reel is not placed in the device when the end of a tape is reached, a message is sent to the operator requesting that the next tape be placed in the device.

FROMRCDLEN

Specifies the length (bytes) of the records contained in the tape from-file. This parameter should be specified in this command or a tape file command for *NS, *NL, *BLP, or *LTM tapes, or for the tapes that do not have HDR2 labels. For *SL tapes, the record length is obtained from the label itself.

***TAPF:** The value specified in the tape device file or override is used.

***CALC:** No record length is specified for the from-file. The system calculates a correct record length. When label processing is bypassed or when the tape from-file does not have standard labels, the record length is calculated from the block length; the record block format must not be blocked or spanned.

record-length: Specify a value, ranging from 1 through 32767, for use as the record length when copying records from the tape from-file. The record length must be consistent with the block length and record block format parameter values.

FROMBLKLEN

Specifies the length (bytes) of data blocks transferred from the tape from-file. This parameter should be specified for nonstandard or nonlabeled tapes, for tapes whose label processing has been bypassed, or for tapes that do not have HDR2 labels. For standard labeled tapes, the block length is obtained from the label itself, and this parameter is ignored.

***TAPF:** The value specified in the tape device file or override is used.

***CALC:** No block length is specified for the from-file. The system calculates a correct block length. When label processing is bypassed or when the tape from-file does not have standard labels, the block length is calculated from the record length. The record block format must not be blocked or spanned.

block-length: Specify a value, ranging from 18 through 32767, that specifies the block length of each block in the tape from-file. The block length must be consistent with the record length and record block format values.

FROMRCDBLK

Specifies the record block format and blocking attribute of records in the tape from-file. This parameter should be specified for nonstandard or nonlabeled tapes, for tapes whose label processing has been bypassed, or for tapes that do not have HDR2 labels. For standard labeled tapes, the record block format is obtained from the HDR2 label itself and this parameter is ignored.

***TAPF:** The value specified in the tape device file or override is used.

***D:** The tape from-file has variable-length, unblocked records in ASCII type D format.

***DB:** The tape from-file has variable-length, blocked records in ASCII type D format.

***F:** The tape from-file has fixed-length, unblocked records in either EBCDIC or ASCII format.

***FB:** The tape from-file has fixed-length, blocked records in either EBCDIC or ASCII format.

***U:** The tape from-file has records in an undefined format.

***V:** The tape from-file has variable-length, unblocked records in EBCDIC format.

***VB:** The tape from-file has variable-length, blocked records in EBCDIC format.

***VBS:** The tape from-file has variable-length, spanned blocked records.

***VS:** The tape from-file has variable-length spanned records.

FROMENDOPT

Specifies the positioning operation that is performed automatically on the last from-file tape volume when the tape device file is closed. For a multi-volume tape from-file, this parameter applies to the last reel (volume) only; all intermediate volumes are copied with *UNLOAD as the positioning attribute.

***TAPF:** The value specified in the tape device file or override is used.

***REWIND:** The tape is automatically rewound, but not unloaded, after the operation has ended.

***UNLOAD:** The tape is automatically rewound and unloaded after the operation ends.

***LEAVE:** The tape does not rewind or unload after the operation ends. It remains at the current position on the tape drive.

MBROPT

Specifies whether the new records replace or are added to the existing records.

Note: This parameter is required for a physical file and ignored for a device to-file.

***NONE:** No records are added or replaced in the member. This value is valid only for a copy to a device file.

***ADD:** The system adds the new records to the end of the existing records.

***REPLACE:** The system clears the existing member and adds the new records.

NBRRCDS

Specifies the number of records copied to the to-file.

***END:** Records are copied until the end-of-file condition is indicated.

number-of-records: Specify the number of records, ranging from 1 to 2147483647, that are copied to the to-file. Fewer records are copied if an end-of-file condition occurs before the specified number of records have been copied.

OUTFMT

Specifies, if TOFILE(*PRINT) is specified, whether the copied records are printed in character or hexadecimal format.

***CHAR:** Records are printed in character format only.

***HEX:** Records are printed in both character format and hexadecimal format.

Examples

Example 1: Adding Copied Records to Existing Records

```
CPYFRMTAP  FROMFILE(QTAPE)  TOFILE(DEPT/YTDSALES)
  FROMSEQNBR(3)  FROMLABEL(DAILY)  FROMDEV(QTAPE1)
  MBROPT(*ADD)
```

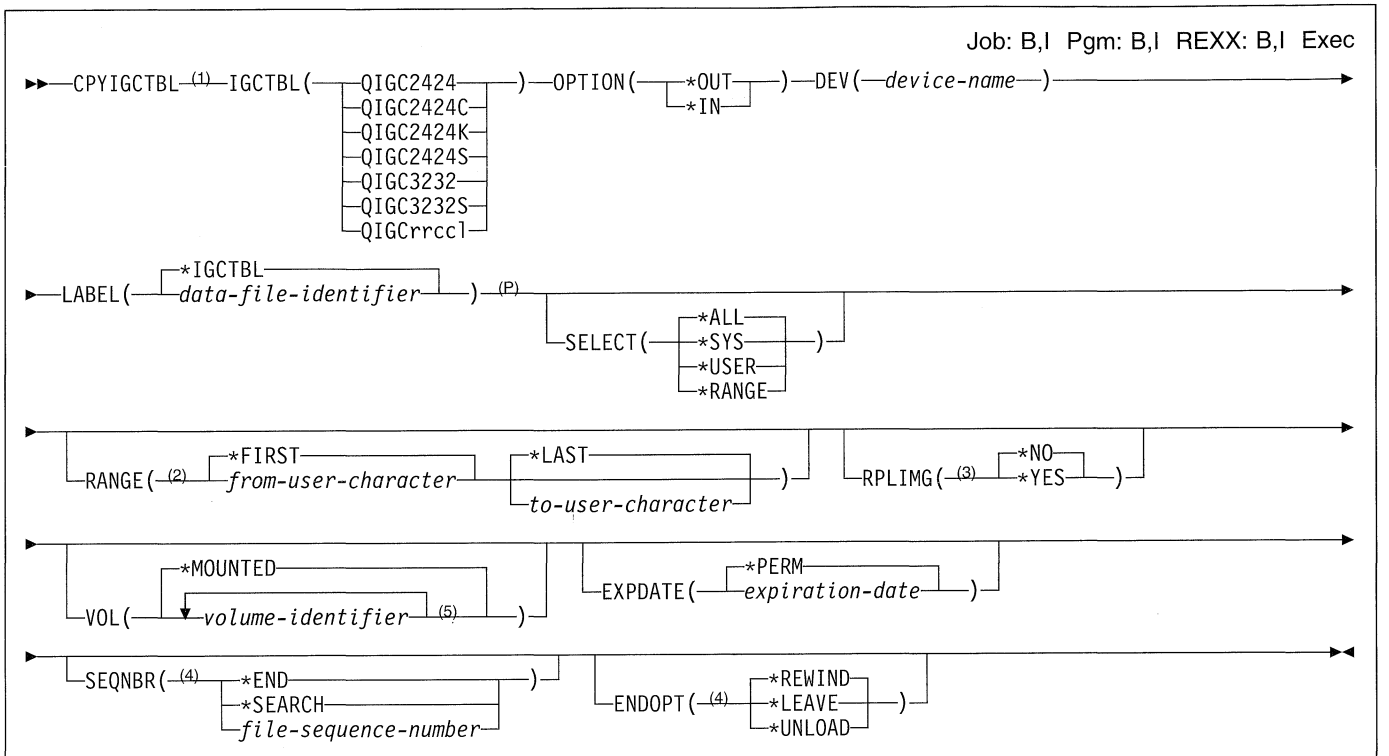
This command copies records from tape by using the tape device file QTAPE. The data file at sequence number 3 labeled DAILY on device QTAPE1 is copied. The specific attributes of the data file, such as record length and record block format, is determined by the system from the label on the tape. The records are added to the existing records in the member DAILY, which is implied by the parameter default of (TOMBR(*FROMLABEL), in file YTDSALES, which is in library DEPT.

Example 2: Replacing Existing Records

```
CPYFRMTAP  FROMFILE(QTAPE)  TOFILE(MYLIB/KEN)
  FROMSEQNBR(2)  FROMDEV(QTAPE1)  FROMREELS(*NL 1)
  FROMRCDLEN(100)  FROMBLKLEN(1000)
  FROMRCDBLK(*FB)  TOMBR(*FIRST)  MBROPT(*REPLACE)
```

This command copies records from tape by using the tape device file QTAPE. Records in the data file at sequence number 2 of a nonlabeled tape file on tape device QTAPE1, with fixed-length records that are blocked 1000 bytes (or 10 records) to a block, are copied, and replace the existing records in the first member in file KEN, which is in library MYLIB.

CPYIGCTBL (Copy DBCS Font Table) Command



Notes:

1 DBCS systems only

2 Range keyword valid only if SELECT(*RANGE) is specified.

3 RPLIMG keyword not valid if SELECT(*SYS) is specified.

4 Applies to tape devices only. Default SEQNBR depends on OPTION VALUE. For OPTION(*OUT), default is SEQNBR(*END). For OPTION(*IN), default is SEQNBR(*SEARCH).

P All parameters preceding this point can be specified in positional form.

5 A maximum of 10 repetitions

Purpose

The Copy DBCS Font Table (CPYIGCTBL) command copies part or all of a double-byte character set (DBCS) font table from system storage to tape or diskette, or from tape or diskette into system storage. Copying a DBCS font table from tape or diskette into system storage also puts its definition in the system. DBCS font tables are objects and can be saved and restored.

DBCS font tables contain the images in a given dot matrix of the DBCS extension characters used on the system. The system refers to the tables when printing and displaying these characters. There are separate tables for each character image matrix used by devices attached to the system.

Required Parameters

IGCTBL

Specifies the name of the DBCS font table being copied. Choose one of the following table names:

QIGC2424: The Japanese DBCS font table used for displaying and printing extension characters in a 24-by-24 dot matrix image.

QIGC2424C: The Traditional Chinese DBCS font table used for printing extension characters in a 24-by-24 dot matrix image.

QIGC2424K: The Korean DBCS font table used for printing extension characters in a 24-by-24 dot matrix image.

QIGC2424S: The Simplified Chinese DBCS font table used for printing extension characters in a 24-by-24 dot matrix image.

QIGC3232: The Japanese DBCS font table used for displaying and printing extension characters in a 32-by-32 dot matrix image.

QIGC3232S: The Simplified Chinese DBCS font table used for printing extension characters in a 32-by-32 dot matrix image.

QIGCrrccl: The name of the DBCS font table to be copied must always be in the format QIGCrrccl, where *rr* is the table row matrix size, *cc* is the table column matrix size, and the letter *l* is an optional language identifier.

OPTION

Specifies how to copy the DBCS font tables: either from the system to diskette or tape, or from diskette or tape into the system.

***OUT:** The specified DBCS font table is copied to diskette or tape.

***IN:** The specified DBCS font table is copied from diskette or tape to the system.

DEV

Specifies the name of the diskette or tape device that the table is saved to or restored from. The device name must already be known on the system by a device description.

LABEL

Specifies the object that identifies the data file on diskette or tape that contains the DBCS font table. When copying the table into the system, the label identifies the file that exists on diskette or tape. When copying the table to diskette or tape, the label identifies the file that is created on diskette or tape.

***IGCTBL:** Identifies the data file on diskette or tape that contains the DBCS font table. It is the same as the name specified on the IGCTBL parameter without the first character.

data-file-identifier: Specify the identifier (8 characters maximum for diskette and 17 maximum for tape, starting with an alphabetic character) of the data file that is used with this DBCS font table.

Optional Parameters

SELECT

Specifies which portion of the DBCS font table is to be copied.

***ALL:** All IBM-supplied and user-defined DBCS characters are copied.

***SYS:** Only IBM-supplied DBCS characters are copied.

***USER:** Only user-defined DBCS characters are copied.

***RANGE:** Only user-defined DBCS characters that fall in the range specified on the RANGE parameter are copied.

RANGE

Specifies the range of user-defined characters to be copied. Specify this value only when choosing SELECT(*RANGE).

The first of two values specified with the RANGE parameter specifies the first character to be copied. Specify one of the following values:

Element 1: Starting Character to be Copied

***FIRST:** The system starts copying with the first user-defined DBCS character in the table.

from-user-character: The system starts copying with the specified DBCS code or number. Specify the 4-character DBCS code, or the 5-character DBCS number.

The last of two values specified with the RANGE parameter specifies the last character to be copied. Specify one of the following values:

Element 2: Ending Character to be Copied

***LAST:** The system stops copying with the last user-defined character found.

to-user-character: The system stops copying with the specified DBCS code or number. Specify the 4-character DBCS code, or the 5-character DBCS number.

RPLIMG

Specifies whether user-defined DBCS characters in the specified table are replaced with those found on tape or diskette. Specify this value only when choosing OPTION(*IN).

***NO:** The system does not replace user-defined DBCS characters in the table stored in the system with those found on tape or diskette.

***YES:** The system replaces user-defined DBCS characters in the table stored in the system with those found on tape or diskette.

VOL

Specifies one or more volume identifiers used by the file. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***MOUNTED:** The volume currently placed in the device is used.

volume-identifier: Specify the identifiers of one or more diskettes or tapes in the order in which they are placed in a tape or diskette drive and used in the copy operation.

EXPDATE

Specifies, when OPTION(*OUT) is indicated, the end date of the file on diskette or tape that contains the DBCS font table. If a date is specified, the file is protected and cannot be written over until the day after the specified end date.

***PERM:** The data file is permanently protected. An expiration date of 999999 is assigned.

expiration-date: Specify the expiration date after which the data file is no longer protected. The date must be specified in the format defined by the system values QDATFMT and, if separators are used, QDATSEP. However, the specified date is put in the diskette or tape label as yymmdd, where yy is the year, mm is the month, and dd is the day.

SEQNBR

Specifies (only when tape is used) which sequence number is used as the starting point for the copy process. This value is valid only if the parameter OPTION(*OUT) is specified.

Note: This value is valid only if *OUT is specified on the OPTION parameter.

***END:** The system saves the table after the last sequence number on the tape.

***SEARCH:** The volume that is in the tape or diskette unit is searched for a data file with an identifier that compares with the LABEL parameter value; when a match is found, the table is restored. If the last operation on the device specifies ENDOPT(*LEAVE) (the tape is positioned at the location at which the last operation ended), the file search starts with the first data file beyond the current tape position. If ENDOPT(*LEAVE) was not used for the last operation (or if the tape was manually rewound since an ENDOPT(*LEAVE) operation), the search starts with the first data file on the volume. This value is valid only if the parameter OPTION(*IN) is specified.

file-sequence-number: Specify the sequence number of the file that is used.

ENDOPT

Specifies the operation that is automatically performed on the tape volume after the operation ends. If more than one volume is included, this parameter applies only to the last tape volume used; all other tape volumes are rewound and unloaded when the end of the tape is reached.

***REWIND:** The tape is automatically rewound, but not unloaded, after the operation has ended.

***LEAVE:** The tape does not rewind or unload after the operation ends. It remains at the current position on the tape drive.

***UNLOAD:** The tape is automatically rewound and unloaded after the operation ends.

Example

```
CPYIGCTBL  IGCTBL(QIGC2424)  OPTION(*OUT)  LABEL(*IGCTBL)
  DEV(QDKT)
```

This command causes the system to copy the complete Japanese DBCS font table QIGC2424 from the system to the diskette. The name of the label on the diskette is IGC2424.

Additional Considerations

Consider the following before entering this command:

- The diskette used in the copy operation must be in the *DATA format. The *Operator's Guide* has instructions on initializing diskettes in the *DATA format.
- The system creates the DBCS font table in addition to copying it when you specify OPTION(*IN), if the following is true:
  - The specified table does not already exist in the system.
  - The tape or diskette that you are copying the table from contains all of the DBCS characters supplied with your system.
  - SELECT(*ALL) or SELECT(*SYS) was specified.
- Consider copying a DBCS font table to tape or diskette before deleting that table from the system.
- The following DBCS codes can be specified as the RANGE value for Japanese use:

6941 - 69FE	6A41 - 6AFE	6B41 - 6BFE
6C41 - 6CFE	6D41 - 6DFE	6E41 - 6EFE
6F41 - 6FFE	7041 - 70FE	7141 - 71FE
7241 - 72FE	7341 - 73FE	7441 - 74FE
7541 - 75FE	7641 - 76FE	7741 - 77FE
7841 - 78FE	7941 - 79FE	7A41 - 7AFE
7B41 - 7BFE	7C41 - 7CFE	7D41 - 7DFE
7E41 - 7EFE	7F41 - 7FFE	

- The following DBCS numbers are specified as the RANGE value for Japanese use:

10561 through 10750	10817 through 11006
11073 through 11262	11329 through 11518
11585 through 11774	11841 through 12030
12097 through 12286	12353 through 12542
12609 through 12798	12865 through 13054
13121 through 13310	13377 through 13566
13633 through 13822	13889 through 14078
14145 through 14334	14401 through 14590
14657 through 14846	14913 through 15102
15169 through 15358	15425 through 15614
15681 through 15870	15937 through 16126
16193 through 16382	

- The following DBCS codes are specified as the RANGE value for Korean use:

D441 - D4FE	D541 - D5FE	D641 - D6FE
D741 - D7FE	D841 - D8FE	D941 - D9FE
DA41 - DAFE	DB41 - DBFE	DC41 - DCFE
DD41 - DDFE		

- The following DBCS numbers are specified as the RANGE value for Korean use:

37953 through 38142	38209 through 38393

38465 through 38654	38721 through 38910
38977 through 39166	39233 through 39422
39489 through 39678	39745 through 39934
40001 through 40190	40257 through 40446

- The following DBCS codes are specified as the RANGE value for Traditional Chinese use:

D041 - D0FE	D141 - D1FE	D241 - D2FE
D341 - D3FE	D441 - D4FE	D541 - D5FE
D641 - D6FE	D741 - D7FE	D841 - D8FE
D941 - D9FE	DA41 - DAFE	DB41 - DBFE
DC41 - DCFE	DD41 - DDFE	

- The following DBCS numbers are specified as the RANGE value for Traditional Chinese use:

36929 through 37118	37185 through 37374
37441 through 37630	37697 through 37886
37953 through 38142	38209 through 38398
38465 through 38654	38721 through 38910
38977 through 39166	39233 through 39422
39489 through 39678	39745 through 39934
40001 through 40190	40257 through 40446

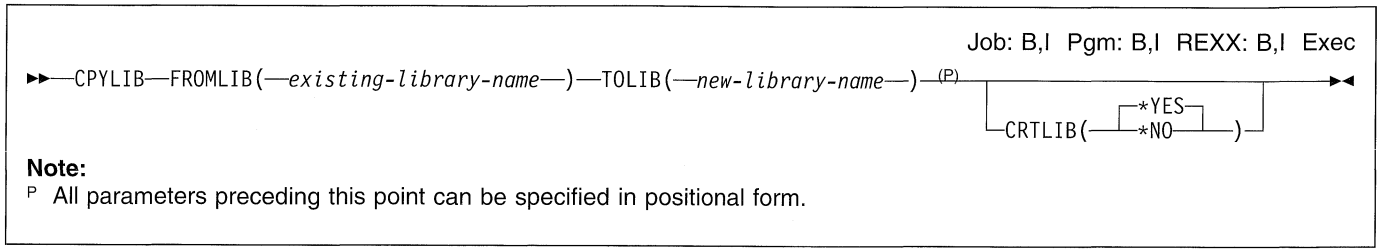
- The following DBCS codes are specified as the RANGE value for Simplified Chinese use:

7641 through 76FE	7741 through 77FE
7841 through 78FE	7941 through 79FE
7A41 through 7AFE	7B41 through 7BFE
7C41 through 7CFE	7D41 through 7DFE
7E41 through 7EFE	7F41 through 7FFE

- The following DBCS numbers are specified as the RANGE value for Simplified Chinese use:

13889 through 14078	14145 through 14334
14401 through 14590	14657 through 14846
14913 through 15102	15169 through 15358
15425 through 15614	15681 through 15870
15937 through 16126	16193 through 16382

CPYLIB (Copy Library) Command



Note:
P All parameters preceding this point can be specified in positional form.

Purpose

The Copy Library (CPYLIB) command copies all of the objects in a library to another library, and creates the TOLIB if it does not exist.

Restrictions:

1. If this command is interrupted prior to completion, the results are unpredictable.
2. Only object types supported by the Create Duplicate Object (CRTDUPOBJ) command can be copied.
3. If another job is processing any of the objects in the library, the results may be unpredictable. For example:
   - Objects created in the library after the copy starts are not copied.
   - If objects are deleted from the library after the copy starts, a user of the CPYLIB gets an error message.
   - If objects are in use or are locked by another job, the copy may not be successful.
4. Other jobs should not attempt to use objects in the new library until the copy is complete.
5. The restrictions for the Create Duplicate Object (CRTDUPOBJ) command also apply to the CPYLIB command.
6. The to library is created in the same ASP (Auxiliary Storage Pool) as the from library.
7. All copied objects are created in the same ASP as the original objects.
8. If objects already exist in the to library when the copy operation is performed, the existing objects will remain in the to library after the new objects are copied.

Required Parameters

FROMLIB
Specifies the name of the library that is copied.

TOLIB
Specifies the name of the new library that is created.

Optional Parameters

CRTLIB
Specifies whether the new library replaces an existing library.

***YES:** The TOLIB is created during the copy from the FROMLIB. If the TOLIB already exists, the CPYLIB ends.

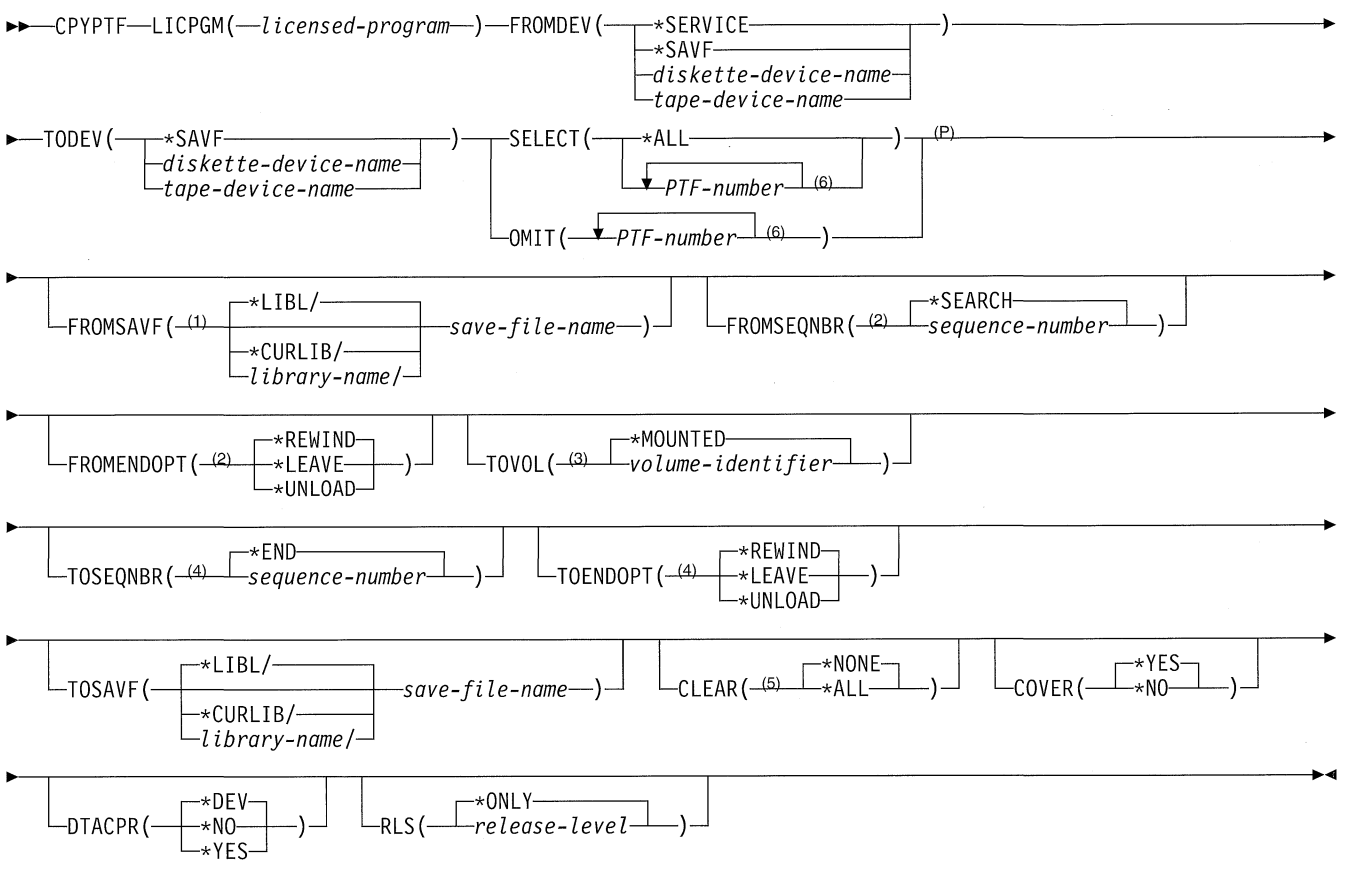
***NO:** The TOLIB is not created (it is assumed to already exist). If the TOLIB does not exist, the CPYLIB ends.

Example

```
CPYLIB   FROMLIB(MYLIB)  TOLIB(NEWLIB)
```

This command copies the contents of library MYLIB into library NEWLIB.

CPYPTF (Copy Program Temporary Fix) Command



Notes:

1 FROMSAVF can only be used with FROMDEV(*SAVF).

2 FROMSEQNBR and FROMENDOPT can only be used when FROMDEV specifies a tape.

3 TOVOL applies only to tape or diskette.

4 TOSEQNBR and TOENDOPT apply only when TODEV is a tape device.

5 CLEAR can be used only when TODEV(*SAVF) is specified.

6 A maximum of 50 repetitions. PTF numbers may not be specified for both SELECT and OMIT.

P All parameters preceding this point can be specified in positional form.

Purpose

The Copy Program Temporary Fix (CPYPTF) command copies program temporary fixes (PTFs) from diskette, tape, service, or save files to a diskette, tape, or a save file. Each PTF contains one or more objects, including programs, which can be loaded by the Load Program Temporary Fix (LODPTF) command or applied to a licensed program by the Apply Program Temporary Fix (APYPTF) command.

Restrictions:

1. This command is shipped with public *EXCLUDE authority and the QPGMR, QSYSOPR, QSRV, and QSRVBAS user profiles have private authorities to use the command.
2. A PTF can be loaded on a system, but can only be copied from that system to the previous release system. For example, a V2R1M0 PTF can be copied to a V2R3M0 system, but if it is then copied to a tape or diskette, it cannot be loaded on a V2R1M0 system. The PTF can be copied from the V2R3M0 system to a V2R2M0 system and then copied to a tape or diskette and then loaded on a V2R1M0 system.
3. When this command is running, error messages CPD6420, CPD6421, or CPF3741 may be issued indicating that the PTF cannot be taken to an earlier release of the operating system. These messages do not mean that the PTF was not copied. PTFs are copied to the earliest release format on which the PTF can be saved. Two or three saves may be attempted, causing error

messages in the job log, before a successful save occurs. If a copy PTF failure message is not issued, a successful copy is performed regardless of the other error messages.

Required Parameters

LICPGM
Specifies the licensed program number of the PTFs to be copied. Specify the number of the licensed program.

FROMDEV
Specifies the name of the device that contains the PTFs to be copied.

***SERVICE:** The PTFs sent from the service support system are copied.

***SAVF:** The PTFs are copied from a save file.

diskette-device-name: Specify the name of the diskette device from which the PTFs are copied.

tape-device-name: Specify the name of the tape device from which the PTFs are copied.

TODEV
Specifies the device to which the PTFs are copied.

***SAVF:** The PTFs are copied to a save file.

diskette-device-name: Specify the diskette device name to which the PTFs are copied.

tape-device-name: Specify the name of the tape device to which the PTFs are copied.

SELECT
Specifies which of the PTFs for the specified licensed program are copied. The OMIT parameter cannot be specified if SELECT is specified with PTF numbers.

***ALL:** All the PTFs for the specified program are copied.

PTF-number: Specify the PTF identification numbers of the PTFs that are copied.

OMIT
Specifies that all PTFs, except for those specified in this parameter, are copied.

PTF-number: Specify the PTF numbers of the PTFs to be omitted. Up to 50 PTF numbers can be specified. The OMIT parameter cannot be specified if single PTF numbers are specified on the SELECT parameter.

Optional Parameters

FROMSAVF
Specifies the save file from which the PTFs are copied.

The name of the save file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

save-file-name: Specify the name of the save file from which the PTFs are copied.

FROMSEQNBR
Specifies the sequence number on the tape volume from which the PTFs are copied.

***SEARCH:** The tape volume is searched for the first program temporary fix (PTF) file for the specified licensed program.

sequence-number: Specify the sequence number at which to begin to copy the PTF data. This sequence number must exist on the tape.

FROMENDOPT
Specifies the operation that is automatically performed on the tape volume after the copy PTF operation ends. If one or more reels of tape is involved, this parameter applies only to the last reel. This parameter is only valid if a tape-device-name is specified on the FROMDEV parameter. This parameter is ignored if a diskette-device-name is specified on the FROMDEV parameter.

***REWIND:** The tape is automatically rewound, but not unloaded, after the operation has ended.

***LEAVE:** The tape does not rewind or unload after the operation ends. It remains at the current position on the tape drive.

***UNLOAD:** The tape is automatically rewound and unloaded after the operation ends.

TOVOL
Specifies the volume identifier of the tape or diskette used when copying.

***MOUNTED:** The volume currently placed in the device is used.

volume-identifier: Specify the volume identifier of the tape or diskette.

TOSEQNBR
Specifies the sequence number of the data file that receives the copied PTFs.

***END:** The system saves the PTFs after the last sequence number on the tape.

sequence-number: Specify the sequence number of the file.

TOENDOPT
Specifies the positioning operation done automatically on the tape volume when the tape device file is closed.

***REWIND:** The tape is automatically rewound, but not unloaded, after the operation has ended.

***LEAVE:** The tape does not rewind or unload after the operation ends. It remains at the current position on the tape drive.

***UNLOAD:** The tape is automatically rewound and unloaded after the operation ends.

TOSAVF
Specifies the qualified name of the save file to which the PTFs are copied.

The name of the save file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

save-file-name: Specify the name of the save file to which the PTFs are copied.

CLEAR
Specifies whether uncleared save files encountered while copying the PTFs are automatically cleared.

***NONE:** An uncleared save file encountered while PTF data is being saved is not automatically cleared.

***ALL:** An uncleared save file, encountered while PTF data is being saved, is automatically cleared so the operation can continue.

COVER
Specifies whether to copy the cover letter from the PTF. This can only be used when the FROMDEV specifies a tape.

***YES:** The cover letter is copied.

***NO:** The cover letter is not copied.

DTACPR
Specifies whether data compression is performed.

Note: Data compression allows the user to customize or create PTF packages and distribute them to other systems on a network.

***DEV:** If the tape device has the hardware compression feature installed, processing proceeds as if DTACPR(*YES) is specified. If the compression feature is not installed or if save data is written to a diskette or save file, processing proceeds as if DTACPR(*NO) is specified.

***NO:** No data compression or decompression occurs.

***YES:** Program data compression is done. If the copy operation is to tape and the target device has the hardware compression feature, hardware compression is done. If the feature is not present or if the copy data is written to a diskette or save file, software compression is done. If the copy operation is running while other jobs on the system are active and software compression is used, overall system performance may be affected.

RLS
Specifies the release level of the software product. If multiple releases are installed, the release is required.

***ONLY:** This value is valid only if *one* release is installed on the system.

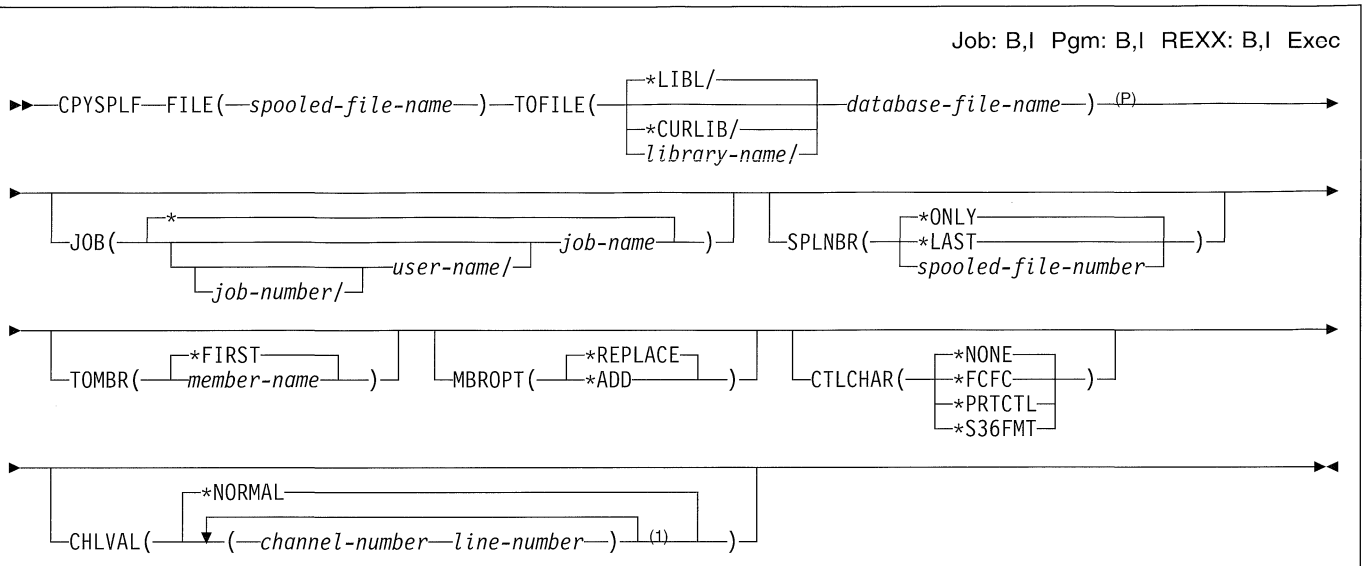
release-level: Specify the release level in the format VxRxMx, where Vx is the version number, Rx is the release number, and Mx is the modification number.

Example

```
CPYPTF  LICPGM(5738SS1)  SELECT(*ALL)  FROMDEV(TAP01)
  TODEV(*SAVF)  TOSAVF(MYLIB/PTFFILE)
```

This command copies all the PTFs from a tape named TAP01 into a save file named PTFFILE for the licensed program 5738-SS1.

CPYSPLF (Copy Spooled File) Command



Notes:

P All parameters preceding this point can be specified in positional form.

1 A maximum of 12 repetitions

Purpose

The Copy Spooled File (CPYSPLF) command copies the data records in the specified spooled file to a user-defined physical data-base file. This conversion allows the use of spooled files in applications using microfiche, data communications, or data processing. Print lines that are all blank are not copied.

When you copy a spooled file to a physical file, certain information is lost or changed, including:

- Graphics data is lost.
- Bar code data is lost.
- Data defined by using the DFNCHR or TRNSPY keywords is replaced with blanks.
- Translation for CHRID and TRNTBL is not performed. (CHRID and TRNTBL are specified through DDS and printer device file parameters.)

If the CPYF command is used to copy the data back to another spooled file, additional information is lost, including:

- For printer files, any attribute that varied within the spooled file. This includes variable CPI, variable LPI, variable FONT, variable CHRID, variable PAGRTT, and variable DRAWER. The entire file is produced with the values specified by the file level attributes. Also, any subscripts, superscripts, or justification of text is lost. Text prints in the default color of the device.
- Any file level attributes (CODE, EXCHTYPE, LABEL, and VOL for diskette files; CHRID, CPI, DEVTYPE, DRAWER, FONT, FORMFEED, PAGESIZE, LPI, PAGRTT, and PRTQLTY for printer files) are set from the device file specified by the TOFILE parameter on the CPYF command. To make the new spooled file look as much like the original spooled file as possible, the attributes of the device file should be overridden to have the same values as the original spooled file. The attributes of the original spooled file can be obtained by using the Work with Spooled File Attributes (WRKSPLFA) command.

Required Parameters

FILE

Specifies the name of the spooled file that is copied to a database file.

TOFILE

Specifies the qualified name of the file that receives the copied records.

Note: If this file does not exist at the time of copy operation is specified, the copy operation fails.

The name of the file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

database-file-name: Specify the name of the database file to which the spooled records are copied.

Optional Parameters

JOB

Specifies the name or qualified name of the job that created the spooled file whose data records are to be copied. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***:** The job that issued this CPYSPLF command is the job that created the spooled file.

If no job qualifier is given, all of the jobs currently in the system are searched for the simple job name.

job-name: Specify the name of the job that created the spooled file.

user-name: Specify the name of the user of the job that created the spooled file.

job-number: Specify the number of the job that created the spooled file.

SPLNBR

Specifies the number of the spooled file from the job whose data records are to be copied. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***ONLY:** One spooled file from the job has the specified file name. The number of the spooled file is not necessary. If *ONLY is specified and more than one spooled file has the specified file name, a message is sent.

***LAST:** The spooled file with the highest number and the specified file name is used.

spooled-file-number: Specify the number of the spooled file whose data records are copied.

TOMBR

Specifies the name of the file member that receives the copied records.

***FIRST:** The first member in the physical file receives the copied records.

member-name: Specify the member name of the physical file. If this member does not exist, a member with the specified name is created and the copy operation continues.

MBROPT

Specifies whether the new records replace or are added to the existing records.

***REPLACE:** The system clears the existing member and adds the new records.

***ADD:** The system adds the new records to the end of the existing records.

CTLCHAR

Specifies to replace the internal print control characters, if any, of a spooled file. Any invalid internal print control characters that are encountered are ignored and formatting may be unpredictable.

***NONE:** No print control characters are created. This option is required for diskette. This option may be useful for microfiche production.

ANSI First-Character Forms-Control Codes

Code	Action before Printing a Line
' '	Space one line (blank code)
0	Space two lines
-	Space three lines
+	Suppress space
1	Skip to next channel 1
2	Skip to next channel 2
3	Skip to next channel 3
4	Skip to next channel 4
5	Skip to next channel 5
6	Skip to next channel 6
7	Skip to next channel 7
8	Skip to next channel 8
9	Skip to next channel 9
A	Skip to next channel 10
B	Skip to next channel 11
C	Skip to next channel 12

***FCFC:** The first character of every record contains an American National Standards Institute (ANSI) forms control character. If *FCFC is specified, the record length must include one extra position for the first-character forms-control code. This value is not valid for externally described printer files.

***PRTCTL:** Specifies that the first four characters of every record contains skip- and space-before values useful in HLL (high-level language) programs. This code can be viewed as SSSL, where SSS is the skip-before line value and L is the space-before value. SSS can range from 001 through 255 to cause a skip to the specified line. Once there, L can be used to specify a spacing of 0, 1, 2, or 3 lines before printing of the record begins. When one part of the code is used (SSS or L), the other part is left blank.

Note: Skip-before line values less than three will not be copied to the to-file; however, they will cause a skip of the specified number of lines. All space-before characters will be copied in the to-file.

Sample control codes and their meanings follow:

Code	Action before Printing a Line
'001 '	Skip to line 1
'010 '	Skip to line 10
'099 '	Skip to line 99
' 1'	Space one line
' 0'	Do not space (or skip)

***S36FMT:** The format of the records copied to a database file is the same as that created by $UASF on the IBM System/36 for COPYPRT. Only spooled printer files can be copied when *S36FMT is specified.

The first record placed in the database file for each spooled file being copied is a header record. Table 21 shows the format for these header records. Columns that are not defined must be blank.

The 2-byte binary numbers are unsigned. That means a page number of 65535 is the highest page number in a header or data record. When the actual page number is higher than 65535, the page numbering wraps beyond 65535 to 0, and then proceeds to 1, 2, 3, and so on.

CHLVAL

Specifies a list of channel numbers with their assigned line numbers. Specify this parameter only if CTLCHAR(*FCFC) has been specified. If the spooled file to be printed has data on a line that immediately precedes a line number assigned to a channel, the copy operation ends.

***NORMAL:** Indicates channel 1 is the only assigned channel number. The assigned line number for channel 1 is line 1.

Element 1: Channel Numbers

channel-number: Specify which ANSI FCFC channels are used to create first-character forms control codes. The only valid values for this parameter range from 1 through 12. Each channel number may be specified only once per CPYSPLF command.

Element 2: Line Numbers

line-number: Specify the line number assigned for the channel number in the same list. Valid line numbers range form 1 through 255. Each line number may be specified only once per CPYSPLF command.

Notes:

1. The order in which the channels are specified on the command is not important. For example, the following lines would be identical:

   ```
   CHLVAL((2  1)(6  15)(8  40))
   CHLVAL((6  15)(2  1)(8  40))
   ```

2. Channel numbers and line numbers do not have to be specified in ascending order.

Table 21. Header Record Format

Beginning Column	Field Length	Contents or Description
1	1	The letter H (to indicate the header)
4	6	The spooled file ID of the entry
12	8	The procedure name
22	8	The job name (the last two characters of the name are truncated)
32	8	The user ID of spooled file creator (the last two characters of the ID are truncated)
42	8	The printer file name (the last two characters of the name are truncated)
52	2	The System/36 printer ID that corresponds to the device that the file prints on. The printer ID shown is the ID for the System/36 environment that copies the file, not the environment that opened the spooled file.
56	4	The forms number (the first 4 characters of the form type of the spooled file)
61	2	The number of copies (in binary)
65	2	The number of pages (in binary)
69	4	The number of data records (in binary) which follow this header record
74	2	The number of lines per page (in binary)
78	1	The letter I if this entry contains print records with double-byte character set (IGC or DBCS) data
81	1	The letter M if this entry contains print records with a length greater than 132
84	1	Lines per inch (in binary)
85	1	Characters per inch (in binary)
86	1	Font ID (in binary)[1]
87	1	Justify[2]
88	1	Align[3]
89	2	The maximum length of the print lines in the spooled file that was copied
92	10	The user name (not truncated) of the creator of the of the spooled file
102	10	The printer file name (not truncated)
112	10	The form type (not truncated)
122	1	The letter Y if this is a copy of a document originally created by DisplayWrite*
123	2	The page width of the DisplayWrite document in 1440ths of an inch (in binary)
125	2	The page length of the DisplayWrite document in 1440ths of an inch (in binary)
127	2	The line increment value (lines per inch) of the DisplayWrite document in 1440ths of an inch (in binary).

1 The IBM AS/400 Font printer file parameter is converted to a binary 1 field. The maximum font ID on IBM System/36 Operation Control Language (OCL) is 255, while IBM AS/400 system supports font identifiers above 255. When an IBM AS/400 system font ID above 255 is used, this field is set to X'0B' for Courier 11 font. If FONT(*CPI) is specified, the field contains X'00'.

2 This is a percent of justification for 5219 printer files. Valid values are: X'00', X'32', and X'64'.

3 Y means to align forms, and N means not to align forms.

The data records placed in the disk file for each copied spooled file have the following format:

Table 22. Copied Spooled File Format

Beginning Column	Field Length	Contents or Description
1	2	The page number (in binary)
3	2	The line number (in binary)
5	4	The record number (in binary)
9	1	The letter I if this print record contains double-byte character set (IGC or DBCS) data
10	1	A double-byte character set (DBCS or IGC) shift-out character (hexadecimal 0E) if this print record starts with DBCS data
11	nnn	The data to be printed[1]

1 The field length is the file record length minus 10. If the print data is too long, it is truncated, and if it is too short, it is padded with blanks. The format of data may not precisely match the format that is produced on IBM System/36 when multiple prints are used to construct a single print line.

Examples

Example 1: Replacing Data

```
CPYSPLF  FILE(QPRINT)  JOB(PAYROLL01)  SPLNBR(4)
  TOFILE(MYFILE)  TOMBR(MYMBR)  CTLCHAR(*PRTCTL)
```

In this example, file QPRINT (which is the fourth file produced by job PAYROLL01) is copied to member MYMBR of physical file MYFILE (which resides in a library found by searching the library list). The newly copied data replaces all old data in the member because all old records have been cleared. The 4-byte print control code is created.

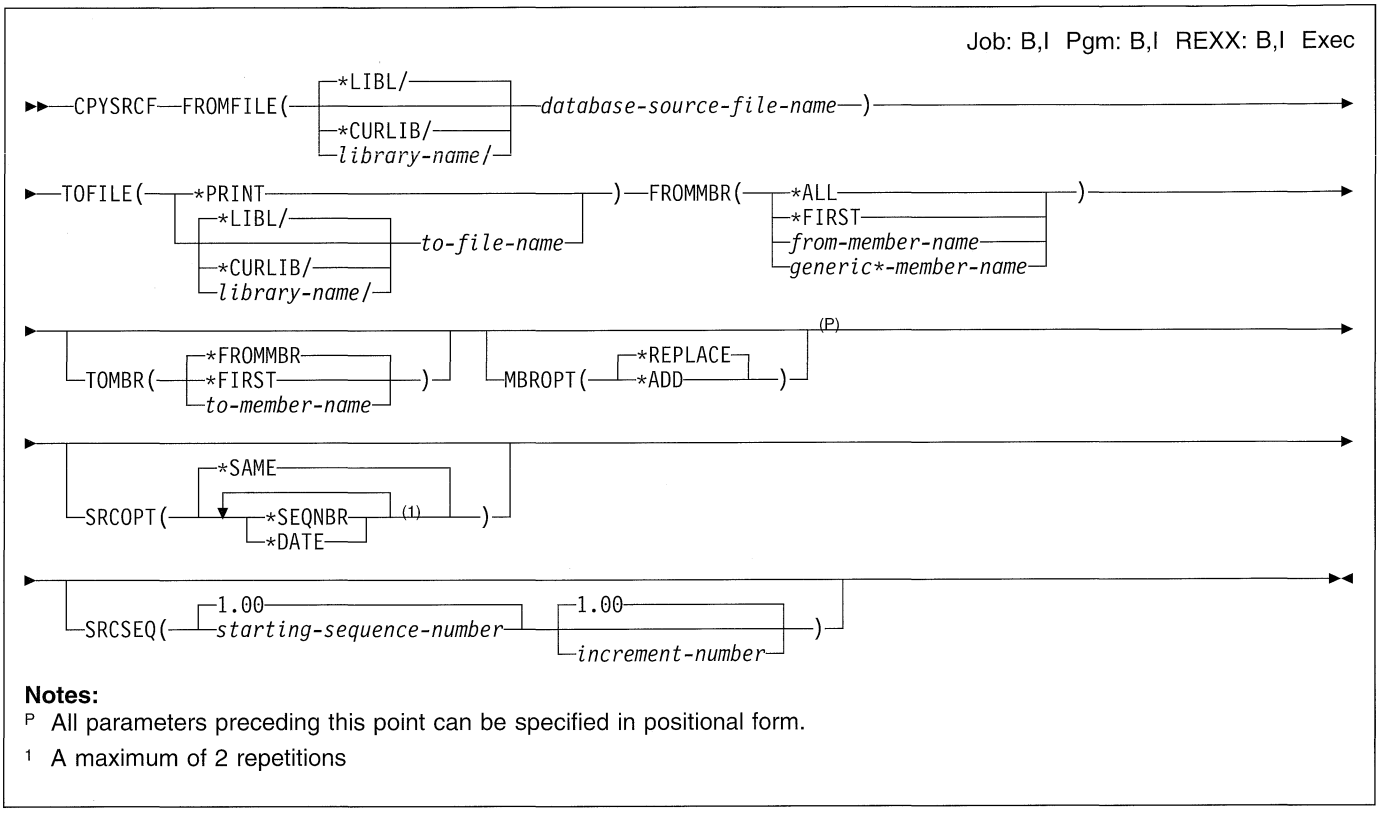
Example 2: Adding Data

```
CPYSPLF  FILE(QPRINT)  TOFILE(MYLIB/MYFILE)
  JOB(PAYROLL02)  MBROPT(*ADD)  CTLCHAR(*FCFC)
  CHLVAL( (1 3) (4  15) )
```

In this example, file QPRINT (the only file of that name left in job PAYROLL02) is copied to the first member of the physical file found in library MYLIB. The newly copied data is added to data existing in the member. The FCFC 1-byte print control character is used and takes advantage of the assigned channel values in formatting the output. The assigned channel values as specified on the command are as follows:

Line 3 assigned to channel 1
Line 15 assigned to channel 4

CPYSRCF (Copy Source File) Command



Notes:

P All parameters preceding this point can be specified in positional form.

1 A maximum of 2 repetitions

Purpose

The Copy Source File (CPYSRCF) command copies a database source file or DDM file to a physical source file or DDM file and converts the data character from the from-file CCSID to the to-file CCSID. If TOFILE(*PRINT) is specified, a formatted printer file is created by using the IBM-supplied printer file QSYSPRT (the file is changed for source records and is different from other copy command file formats). Any overrides issued for the from-file or to-file apply to the files used in the copy operation. Record data is copied from the from-file to the to-file, converting character data from the from-file CCSID to the to-file CCSID. Other differences in record formats (like that of the FMTOPT(*NOCHK) parameter option on the CPYF command) are disregarded.

Note: For more information on DDM files, see the *DDM Guide*.

One member, all members, or a generic set of members can be copied each time the command is called. From-file members can be copied to like-named to-file members or to a single to-file member. Many members are copied and listed in alphabetical order. The to-file must exist when the CPYSRCF command is started. This command does not create the to-file, but it does add a member to an existing physical file if the member does not already exist in the to-file.

This command offers a subset of the parameters available on the CPYF command. Note that the default for the MBROPT parameter is *REPLACE (unlike other copy commands), which clears existing records in the receiving member of the to-file before replacing them with records copied from the from-file. Also, the default for the TOMBR parameter is *FROMMBR, which causes from-file members to be copied to like-named to-file members.

Restriction: A file's open data path (ODP) cannot be shared with any other program in the job (routing step) during the copy operation.

Required Parameters

FROMFILE

Specifies the qualified name of the database source file that contains the records being copied.

The name of the database source file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

database-source-file-name: Specify the name of the database source file that contains records being copied.

TOFILE

Specifies the qualified name of the file that receives the copied records.

***PRINT:** The records are copied to the IBM-supplied printer file QSYSPRT and listed in an SEU-type source file format. No CCSID conversions occur if *PRINT is specified. The format includes no blank lines between records, source fields separated from the data, and members listed in alphabetic order. If the listing needs to be in hexadecimal format, use the Copy File (CPYF) command with the OUTFMT(*HEX) parameter value. The IBM-supplied printer file QSYSPRT may not be overridden to a different file name, and it must have the RPLUNPRT(*YES) and CTLCHAR(*NONE) attributes.

The name of the to-file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

to-file-name: Specify the name of the physical source file that receives the copied records. (If no library qualifier is given, *LIBL is used to find the file.)

FROMMBR

Specifies the members copied from the from-file. A single member, a generic set of members, or all members in the from-file are copied. Members are copied and listed in alphabetic order.

***ALL:** All members in a database file are copied.

***FIRST:** The first member (by creation date) in the database file is copied.

from-member-name: Specify the name of the database file member that is copied.

generic-member-name:* Specify the generic name of the group of members that are copied. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be copied only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

Optional Parameters

TOMBR

Specifies the name of the file member that receives the copied records.

Note: If a member (specified by a name or implied by a parameter value of *FROMMBR) does not exist in the to-file, it is added to the to-file by the copy operation. Either *FROMMBR or *FIRST must be specified for this parameter if the to-file is *PRINT.

***FROMMBR:** The members specified by the FROMMBR parameter are copied into corresponding members in the to-file. If a member with a corresponding name does not exist in the to-file, a member with that name is added to the to-file.

If a member name or *FIRST was specified as a value for the FROMMBR parameter, then a member in the to-file with the same name receives the records copied. If *ALL or a generic member name is specified as a value for the FROMMBR parameter, each member in the from-file is copied into a member with the same name in the to-file. Records from one or more members (specified by the FROMMBR parameter) in the from-file are copied to the first member in the to-file.

***FIRST:** The first member in the database to-file receives the copied records.

to-member-name: Specify the name of the member that receives the records copied from the from-file. If a member with the specified name does not exist, one with the same name is added.

MBROPT

Specifies whether the new records replace or are added to the existing records.

***REPLACE:** The system clears the existing member and adds the new records.

***ADD:** The system adds the new records to the end of the existing records.

SRCOPT

Specifies whether new values are assigned to the source sequence number and date fields when records from the from-file are copied to the to-file. New values can be specified in either or both fields.

***SAME:** The value does not change.

***SEQNBR:** Sequence number and increment values for the sequence number field in the records being copied are assigned as specified in the SRCSEQ parameter.

***DATE:** The date fields in the records being copied are reset to six zeros.

SRCSEQ

Specifies the starting and increment values used for creating sequence numbers for the source sequence number field in the records copied. The maximum value for sequence number is 9999.99; if a value is larger than

this limit, additional records in the member are assigned the sequence number 9999.99. The SRCSEQ parameter is ignored unless SRCOPT(*SEQNBR) is specified.

Element 1: Starting Value

1.00: Specifies that the records being copied to the database file start with a first sequence number of 1.00, and are numbered in increments of 1.00.

starting-sequence-number: Specify the value assigned to the sequence number field for the first record copied. Any value within the range of 0.01 to 9999.99 may be specified.

Element 2: Increment Value

1.00: Specifies that the records being copied to the database file start with a first sequence number of 1.00, and are numbered in increments of 1.00.

increment-number: Specify the value by which the sequence number is incremented for each of the following records copied. Any value in the range of 0.01 to 9999.99 may be specified. Once the maximum sequence number of 9999.99 is reached, the sequence number of additional records is 9999.99.

Examples

Example 1: Replacing Existing Records

```
CPYSRCF  FROMFILE(QGPL/QCLSRC)  TOFILE(MYLIB/CLSRC)
  FROMMBR(PGMA)
```

This command copies records from member PGMA of database source file QCLSRC which is in the QGPL library. The defaults for the TOMBR and MBROPT parameters are taken so the records are copied to a like-named member (PGMA) of CLSRC in library MYLIB and replaces existing records in the member. If member PGMA does not exist in the to-file, it is added as part of the copy operation. If the CCSID of QGPL/QCLSRC is different from the CCSID of MYLIB/CLSRC, the character data is converted to the CCSID of CLSRC.

Example 2: Printing Files

```
CPYSRCF  FROMFILE(QRPG/QRPGSRC)  TOFILE(*PRINT)
  FROMMBR(INV*)
```

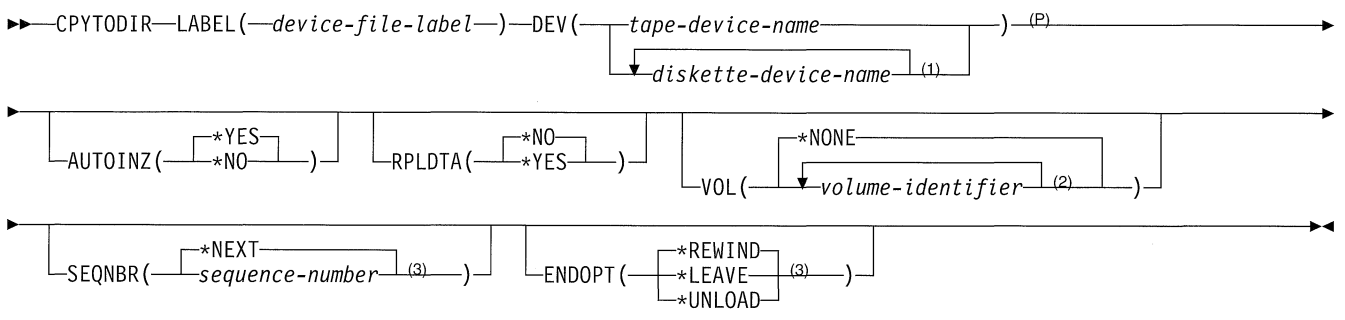
This command copies from database source file QRPGSRC in library QRPG, all file members whose names start with the characters INV. Special value *PRINT is specified for the to-file, so the records are copied to the printer and listed in a format tailored to source records, much like the printout created by SEU. Character data is not converted when specifying TOFILE(*PRINT).

Example 3: Changing the Increment Value

```
CPYSRCF  FROMFILE(MYLIB/TXTSRC)
  TOFILE(QIDU/QTXTSRC)  FROMMBR(*ALL)
  SRCOPT(*SEQNBR *DATE)  SRCSEQ( 1   .25)
```

This command copies all the members of database source file TXTSRC in library MYLIB. They are copied and replace (by using the default MBROPT(*REPLACE)) the existing records in like-named members (by using default TOMBR(*FROMMBR)) of data source file QTXTSRC in library QIDU. If the to-file members do not exist, they are added by the copy operation. For each member copied, the first record is numbered 1 and each following number is incremented by 0.25. Also, the source date field is set to zero in each record. If the CCSID of MYLIB/TXTSRC is different from the CCSID of QIDU/QTXTSRC, the character data is converted to the CCSID of QIDU/QTXTSRC.

CPYTODIR (Copy To Directory) Command



Notes:
1 A maximum of four repetitions.
2 A maximum of 50 repetitions.
3 Applies to tape devices only.
P All parameters preceding this point can be specified in positional form.

Purpose

The Copy To Directory (CPYTODIR) command is used to copy directory data from a tape or diskette device to the local system. The directory data that is copied to the local system is created by the Copy From Directory (CPYFRMDIR) command on a remote system. This function allows the local system to begin a directory shadowing environment with the remote system by shadowing changes made to the directory data from the remote system.

Caution: Do not use this command as a backup utility to save and restore directory data for data recovery purposes. Follow the normal backup and recovery procedure guidelines described in the *Advanced Backup and Recovery Guide*.

Restriction: You must have security administrator (*SECADM) authority to use this command.

Required Parameters

LABEL
Specifies the device file label on the tape or diskette used for the copy operation. A maximum of 17 characters can be specified for tape devices; a maximum of 8 characters can be specified for diskette devices.

DEV
Specifies the names of the tape or diskette devices used for the copy operation. Each tape or diskette device name must already be known on the system by a device description.

tape-device-name: Specify the names of one or more tape devices used for the copy operation. If more than one tape device is used, specify the names of the devices in the order in which they are used. Using more than one tape device permits one tape volume to be rewound and unloaded while another tape device processes the next tape volume.

diskette-device-name: Specify the name of the diskette device to be used for the copy operation.

Optional Parameters

AUTOINZ
Specifies whether to automatically pass the information copied to this system to other collector systems through directory shadowing.

***YES:** The directory entries copied are automatically supplied to collector systems.

***NO:** The directory entries copied are not supplied to other collector systems that have already been initialized.

RPLDTA
Specifies whether to replace a directory entry on the local system with directory data copied from tape or diskette.

***NO:** Shadowed data is copied from tape or diskette, but directory entries are not replaced with copied directory data if the directory entry exists on the system.

***YES:** All directory data is copied from tape or diskette. A directory entry replaces an entry for the same user that already exists on the system.

VOL
Specifies one or more volume identifiers used by the file. More information on this parameter is in Appendix A, "Expanded Parameter Descriptions."

***NONE:** No volume identifiers are specified for the file. No volume identifiers are checked.

volume-identifier: Specify the identifiers of one or more volumes in the order in which they are placed in a device and used for the copy operation.

SEQNBR

Specifies the sequence number of the data file on the tape that is processed. The four-position file sequence number is read from the first header label of the data file.

***NEXT:** The copy operation begins on the next file on the tape volume. If the tape is currently positioned before the first file, the first file from the tape is processed.

file-sequence-number: Specify the sequence number of the file that is used. Valid values range from 1 through 9999.

ENDOPT

Specifies the operation that is automatically performed on the tape volume after the operation ends. If more than one volume is included, this parameter applies only to the last tape volume used; all other tape volumes are rewound and unloaded when the end of the tape is reached.

***REWIND:** The tape is automatically rewound, but not unloaded, after the operation has ended.

***LEAVE:** The tape does not rewind or unload after the operation ends. It remains at the current position on the tape drive.

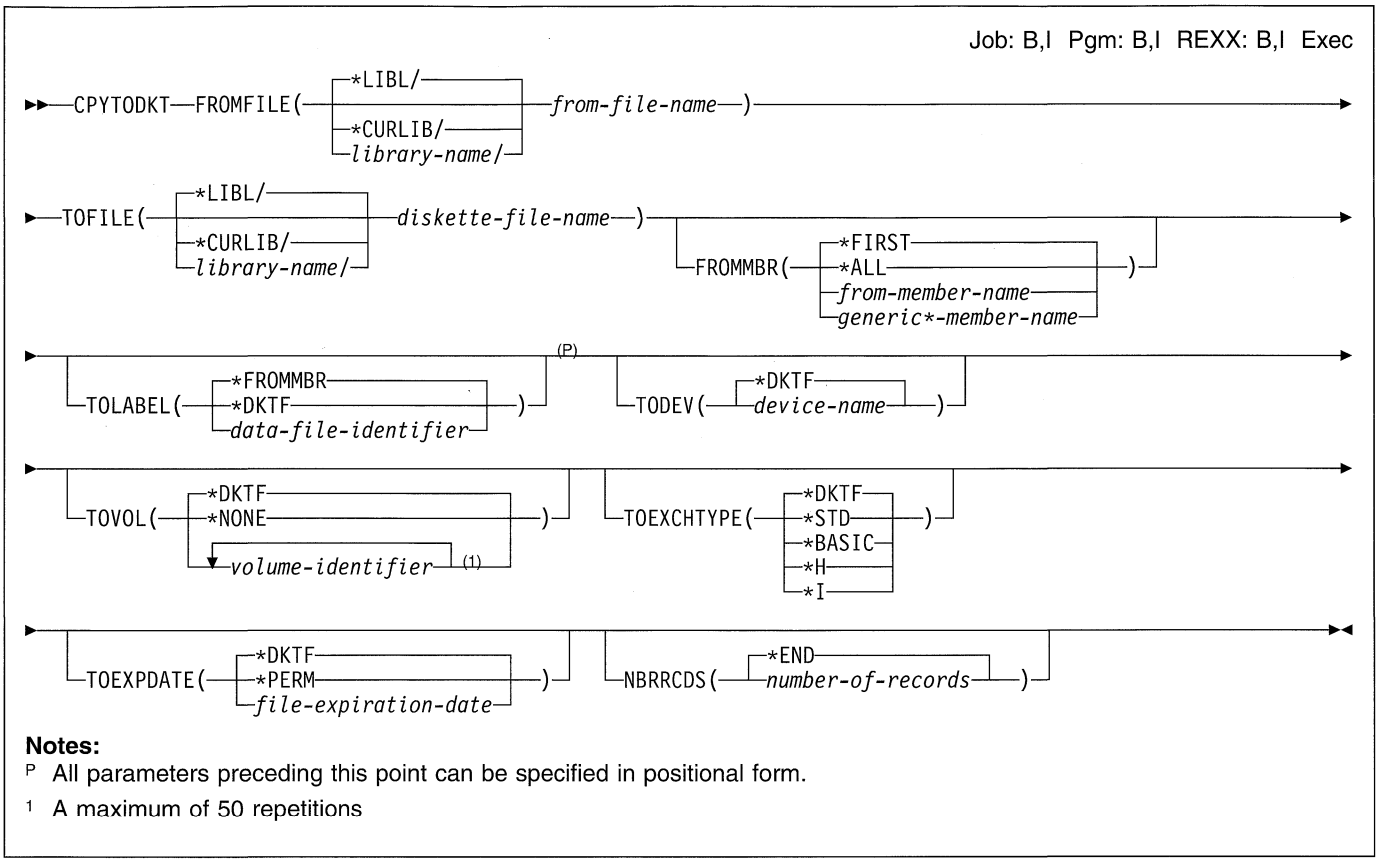
***UNLOAD:** The tape is automatically rewound and unloaded after the operation ends.

Example

```
CPYTODIR   DEV(DKT01)  AUTOINZ(*YES)
  RPLDTA(*NO)
```

This command copies all of the directory data from diskette device DKT01 to the local system. Data that is copied to the local system is passed along to other systems that use communications to perform directory shadow initializing or normal directory shadowing. Local system data identified as being the same as diskette data is not replaced.

CPYTODKT (Copy to Diskette) Command



Notes:

P All parameters preceding this point can be specified in positional form.

1 A maximum of 50 repetitions

Purpose

The Copy to Diskette (CPYTODKT) command copies a database or device file to a diskette file. The to-file must be a diskette file. The from-file can be a physical, logical, DDM, tape, or diskette file, or a spooled inline file.

Note: For more information on DDM files, see the *DDM Guide*.

This command offers a subset of the parameters available on the Copy File (CPYF) command, along with more specific diskette-oriented parameters. This should eliminate the need for a separate override command. If parameters are needed that are not available on the CPYTODKT command, either use overrides for the from-file or the to-file, use the CPYF command, or use a combination of file overrides with the CPYF command.

One label, a generic set, or all labels from the tape can be copied. A complete description of the combinations allowed and how to specify them is in the *Data Management Guide*.

Restriction: A file's open data path (ODP) cannot be shared with any other program in the job (routing step) during the copy operation.

Required Parameters

FROMFILE

Specifies the qualified name of the physical, logical, diskette, tape, or spooled inline file that is copied.

The name of the from-file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

from-file-name: Specify the name of the file being copied.

TOFILE

Specifies the qualified name of the file that receives the copied records.

The name of the diskette file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

diskette-file-name: Specify the name of the diskette file that receives the copied records.

Optional Parameters

FROMMBR

Specifies the database member names, diskette file label identifiers, or a tape file label identifier of records copied from the from-file. If the from-file is a spooled inline file, *FIRST must be specified.

***FIRST:** The first member in the database file is copied. If the from-file is a tape or diskette file, the label in the device file description is used to identify the member being copied.

***ALL:** All members in a database file or all data files on a diskette device file are copied. This value is not valid for a tape file or a spooled inline file.

from-member-name: Specify the name of the database file member or the label of a tape or diskette data file that is copied. If the tape label identifier contains special characters, or is greater than 10 characters in length, it must be specified on the Create Tape File (CRTTAPF), Change Tape File (CHGTAPF), or Override Tape File (OVRTAPF) command.

generic-member-name:* Specify the generic name of the member name. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be copied only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

TOLABEL

Specifies the label identifiers of the diskette data files that receives the records copied from the from-file. The label identifier can be no longer than 8 characters.

***FROMMBR:** Single or many members or label identifiers named in the FROMMBR parameter are copied into data files with the same names in the diskette to-file.

If the from-file is a nonlabeled tape file, a label identifier is created for the diskette to-file in the form of CPYnnnnn, where nnnnn is the tape sequence number of the data file.

***DKTF:** The value in the diskette device file identifies the data file that receives the copied records.

data-file-identifier: Specify the label identifier of the data file on the diskette to-file that receives the records copied from the from-file.

TODEV

Specifies the name of diskette device that is used to copy records to the diskette to-file.

***DKTF:** The devices specified in the diskette device file description are used.

device-name: Specify the names of the diskette device used when writing records to the diskette to-file.

TOVOL

Specifies the diskettes used when copying records to the diskette to-file.

***DKTF:** The diskette volume identifiers in the diskette device file are used to identify the diskettes that receive the copied records.

***NONE:** No volume identifier checking is done.

volume-identifier: Specify up to 50 volume identifiers used to identify the diskettes that receive the copied records. Each volume identifier can have up to 6 alpha-numeric characters.

TOEXCHTYPE

Specifies the exchange type in which the diskette to-file is written.

***DKTF:** The value specified in the diskette device file is used to determine the exchange type of the file being written.

***STD:** The basic exchange format is used for a type 1 or a type 2 diskette. The H exchange type is used for a type 2D diskette.

***BASIC:** The basic exchange type is used.

***H:** The diskette to-file is written in H exchange format. This value is valid for type 2D diskettes initialized with 256-byte sectors.

***I:** The diskette to-file is written in I exchange format. This value is valid for diskette types 1, 2, or 2D, of any valid sector size (128, 256, 512 or 1024 bytes).

TOEXPDATE

Specifies the expiration date for the diskette to-file. If a date is specified, the file is protected and is not overwritten until the day after the expiration date.

***DKTF:** The expiration date specified in the diskette device file is used.

***PERM:** The data file is permanently protected. An expiration date of 999999 is assigned.

file-expiration-date: Specify the date of the last day the diskette is protected. The date must be specified in the format defined by the system values QDATFMT and, if separators are used, QDATSEP. The specified date is put in the diskette label as YYMMDD.

NBRRCDS

Specifies the number of records copied to the to-file.

***END:** Records are copied until the end-of-file condition is indicated.

number-of-records: Specify the number of records, ranging from 1 to 2147483647, that are copied to the to-file. Fewer records are copied if an end-of-file condition occurs before the specified number of records have been copied.

Examples

Example 1: Copying 50 Records

```
CPYTODKT   FROMFILE(PGMSRC)  TOFILE(QDKT)
  FROMMBR(PGMA)  TOLABEL(A)
  TOEXCHTYPE(*I)  NBRRCDS(50)
```

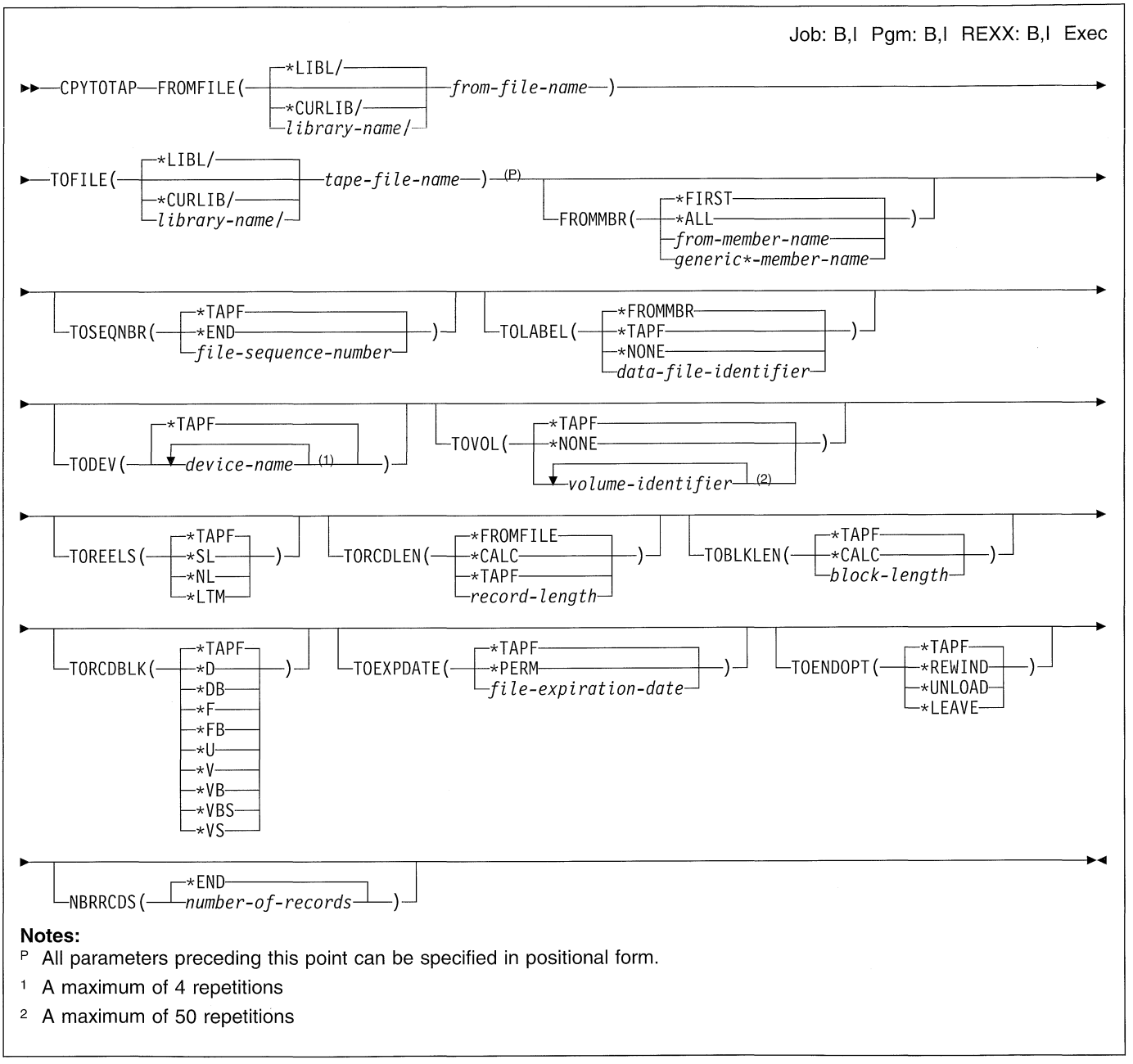
This command copies 50 records to diskette using device file QDKT. The diskette device specified on the QDKTF file description is used. Member PGMA in database source file PGMSRC is copied. Since a library qualifier is not specified, *LIBL is used to find the from-file. The records are copied to label A and are written in I exchange format.

Example 2: Copying All Members

```
CPYTODKT   FROMFILE(QGPL/QCLSRC)  TOFILE(QDKT)
  TODEV(DKT2)  FROMMBR(*ALL)  TOEXPDATE(*PERM)
```

This command copies all the members in database source file QCLSRC in the QGPL library to diskette. The diskette device file QDKT, which is defined as FILETYPE(*DATA), is used even though source records are copied to diskette. This is because source fields are always truncated when copying to a device and appended when copying back to the data base, and using a device file defined as file type *DATA is more efficient than a device file defined as *SRC. The diskette device DKT2 is used. The members are copied to like-named labels on the diskette because the default TOLABEL(*FROMMBR) is taken. The volume and file expiration date are specified on the command, thus eliminating the need for a separate override command.

CPYTOTAP (Copy to Tape) Command



Notes:

P All parameters preceding this point can be specified in positional form.

1 A maximum of 4 repetitions

2 A maximum of 50 repetitions

Purpose

The Copy to Tape (CPYTOTAP) command copies records to a tape file. The to-file must be a tape file. The from-file can be a physical, logical, DDM, diskette, tape, or spooled inline file.

Note: For more information on DDM files, see the *DDM Guide*.

This command offers a subset of the parameters available on the Copy File (CPYF) command, along with more specific tape-oriented parameters. This should eliminate the need for using a separate override command. If parameters that are not available on the CPYTOTAP command are needed, use overrides for the from-file or the to-file, use the CPYF command, or use a combination of file overrides with the CPYF command.

Restrictions:

1. A file's open data path (ODP) cannot be shared with any other program in the job (routing step) during the copy operation.
2. Non-labeled tapes cannot be duplicated to 1/4 inch or 8mm cartridge devices.

Required Parameters

FROMFILE

Specifies the qualified name of the physical, logical, diskette, tape, or spooled inline file from which records are copied.

The name of the from-file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

from-file-name: Specify the file from which records are copied.

TOFILE

Specifies the qualified name of the file that receives the copied records.

The name of the to-file can be qualified by one of the following library values:

***LIBL:** All libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library to be searched.

tape-file-name: Specify the name of the tape device file that receives the copied records.

Optional Parameters

FROMMBR

Specifies, the database file members, diskette data file member identifiers, or a tape file members copied from the from-file. If the from-file is a spooled inline file, *FIRST must be specified for this parameter.

***FIRST:** The first member in the database file is copied. If the from-file is a tape or diskette file, the label in the device file description is used.

***ALL:** All members in a database diskette file are copied.

from-member-name: Specify the name of the database file member or the label of a tape or diskette data file identifier that is copied. If the tape label identifier of the tape or diskette file that is copied contains special characters or is greater than 10 characters in length, then it must be specified on the CRTTAPF, CHGTAPF, or OVRTAPF command for the device file.

generic-member-name:* Specify the generic name of the member. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be copied only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic functions, refer to "Rules for Specifying Names."

TOSEQNBR

Specifies the sequence number of the data file that receives the copied records.

***TAPF:** The value specified in the tape device file is used.

***END:** The system saves the file after the last sequence number on the tape.

For a labeled tape file, the label specified on the TOLABEL parameter must be found at this sequence number. Valid sequence numbers range from 1 through 9999.

If a new data file is added to the tape to-file, the sequence number must be one higher than the sequence number of the last data file on that volume. No gaps are allowed in the series of sequence numbers. If a sequence number of an existing data file is specified, that file is overwritten, and all following files on the volume are destroyed.

file-sequence-number: Specify the sequence number of the file that is used.

TOLABEL

Specifies the label of the data file that receives the copied records. The data file identifier is ignored for a nonlabeled tape.

***FROMMBR:** The receiving data file in the to-file has the same label as the member/label copied from the from-file. If a member/label name was specified as a value for the FROMMBR parameter (or implied by (*TAPF)), then a data-file with the same name in the tape to-file receives the copied records. If a generic name or *ALL is specified as a value for the FROMMBR parameter, then this value is not valid.

If the from-file is a nonlabeled tape file, a label identifier is created for the tape to-file in the form of CPYnnnnn, where nnnnn is the tape sequence number of the data file.

***TAPF:** The value specified in the tape device file or override is used.

***NONE:** No data file identifier is specified. *NONE is not valid if to-file tape has standard labels.

data-file-identifier: Specify the label value (17 characters maximum) that identifies the data file to receive the copied records.

TODEV

Specifies the names of devices that are used to copy records to the tape to-file.

***TAPF:** The value specified in the tape device file is used.

device-name: Specify the device description name or the type of device to use with this communications device entry. The name specified on the CRTDEVxxx command associated with this device description name is used.

TOVOL

Specifies, for tapes with standard labels, one or more volume identifiers of the tapes used when copying records to the tape to-file. The tape volumes must be in the device in the same order as the identifiers are specified. A message is sent to the operator if they are not.

***TAPF:** The value specified in the tape device file is used.

***NONE:** No tape volume identifiers are specified for this file. No volume checking is done beyond verifying that a volume with the correct label type is on the device.

volume-identifier: Specify up to 50 volume identifiers used to identify the tapes to receive the copied records. The tape volumes must be on the device in the same order as the identifiers are specified and in the same order as the device names are specified on the TODEV parameter. Each identifier can have up to 6 alphanumeric characters.

TOREELS

Specifies the type of labeling used on the tape reels that receive the copied records. The system writes only to standard-label tape or to nonlabeled tapes.

***TAPF:** The value specified in the tape device file is used.

***SL:** The volume used for receiving copied records has standard labels.

***NL:** The volume used for receiving copied records has no labels. Tape marks are used to indicate the end of each volume and the data file on it.

***LTM:** The volume used for receiving copied records has no labels, but has a leading tape mark before the first data record.

TORCDLEN

Specifies the length (in bytes) of the records written on the tape to-file. The records in the from-file may be truncated or padded with blanks to conform to the record length specified before they are written to the tape. If an existing data file (with standard labels) is being extended on tape, the record length is obtained from the second header label (HDR2). The record length, block length and record block format must be consistently specified. If a new data file is being added or if it is a nonlabeled file, the record length must be explicitly specified by this parameter or by the RCDLEN parameter in one of the tape device file commands.

***FROMFILE:** The record length for the to-file is the same as the record length of the from-file. If the from-file has a variable record length, the maximum record length of the from-file is used as the record length of the to-file.

***CALC:** No record length is specified for the tape to-file. The record length from the existing data file label is used if an existing data file is being extended in the to-file.

***TAPF:** The value specified in the tape device file is used.

record-length: Specify the record length, ranging from 1 through 32767 bytes, of records that are written to the to-file. The record length must be consistent with the block length and the record block format values.

TOBLKLEN

Specifies the length of data (in bytes) of data blocks transferred to the tape. The block length must be consistent with the record length and record block format values.

***TAPF:** The value specified in the tape device file is used.

***CALC:** No block length is specified for the tape to-file. The block length from the existing data file label is used if the copied records are being written into a data file that is being extended.

block-length: Specify the maximum length of the data block written to the tape. The minimum block size value is 18 bytes and the maximum size is 32767 bytes.

TORCDBLK

Specifies the record format and blocking attributes of records being copied to the tape to-file. The record block format from an existing data file label is used if an existing data file is being extended as the to-file. The record length, block length, and record block format parameters must be consistently specified.

***TAPF:** The value specified in the tape device file is used.

***D:** The tape to-file has variable-length, unblocked records in ASCII type D format.

***DB:** The tape to-file has variable-length, blocked records in ASCII type D format.

***F:** The tape to-file has fixed-length, unblocked records in either EBCDIC or ASCII format.

***FB:** The tape to-file has fixed-length, blocked records in either EBCDIC or ASCII format.

***U:** The tape to-file has records in an undefined format.

***V:** The tape to-file has variable-length, unblocked records in EBCDIC format.

***VB:** The tape to-file has variable-length, blocked records in EBCDIC format.

***VBS:** The tape to-file has variable, spanned blocked records.

***VS:** The tape to-file has variable, spanned records.

TOEXPDATE

Specifies the expiration date for the data file being added to the to-file. The data file expiration date is established for standard label tape files only and is stored in the tape header label immediately preceding the data file that the label describes.

***TAPF:** The value specified in the tape device file is used.

***PERM:** The data file is permanently protected. An expiration date of 999999 is assigned.

file-expiration-date: Specify the expiration date that is used for the data file. The date must be specified in the format defined by the system values QDATFMT and, if separators are used, QDATSEP.

TOENDOPT

Specifies the positioning operation that is done automatically on the to-file tape volume when the tape device-file is closed. For a multivolume tape to-file, this parameter applies to the last reel only; all intermediate reels are positioned *UNLOAD.

***TAPF:** The value specified in the tape device file is used.

***REWIND:** The tape is automatically rewound, but not unloaded, after the operation has ended.

***UNLOAD:** The tape is automatically rewound and unloaded after the operation ends.

***LEAVE:** The tape does not rewind or unload after the operation ends. It remains at the current position on the tape drive.

NBRRCDS

Specifies the number of records copied to the to-file.

***END:** Records are copied until the end-of-file condition is indicated.

number-of-records: Specify the number of records, ranging from 1 to 2147483647, that are copied to the to-file. Fewer records are copied if an end-of-file condition occurs before the specified number of records have been copied.

Examples

Example 1: Copying the First Member in a Database File to Tape

```
CPYTOTAP   FROMFILE(MYLIB/MYFILE)  TOFILE(QTAPE)
  TODEV(QTAPE1)   TORCDLEN(*FROMFILE)
  TOEXPDATE(10 15 89)
```

This command copies the first member in database file MYFILE in library MYLIB to tape. The tape device file QTAPE is used which has attributes of SEQNBR(1) and RCDBLKFMT(*F), which are used as defaults for parameters TOSEQNBR and TORCDBLK. The additional attributes specified on the copy command include the tape device name and the expiration date of the file written on tape. The label name used on tape is the same name as that of the member copied. The record length of the tape file is also the same as the database from-file record length since TORCDLEN (*FROMFILE) was specified.

Example 2: Copying a Member in a Database File to Tape

```
CPYTOTAP   FROMFILE(MYLIB/MYFILE)  TOFILE(QTAPE)
  FROMMBR(M1)  TOLABEL(BACKUPM1)  TODEV(QTAPE1)
  TORCDLEN(50)  TOBLKLEN(1000)  TORCKBLK(*FB)
  TOENDOPT(*LEAVE)
```

This command copies member M1 from database file MYFILE in library MYLIB to tape, using tape device file QTAPE. The records are copied to label BACKUPM1 on tape device QTAPE1 and the reel is left at its current position when the copy is completed. The data file written on tape has fixed-length 50-byte records that are blocked 1000 bytes (or 20 records) to a block.

Program Number: 5738-SS1

Printed in Denmark by Bonde's

SC41-0030-02