

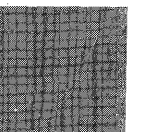
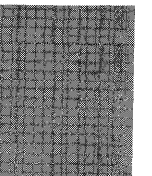
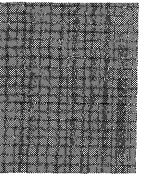
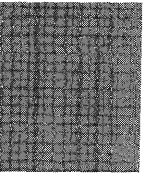
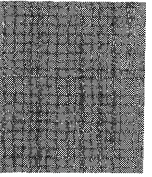
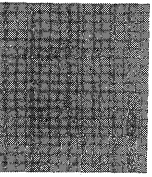
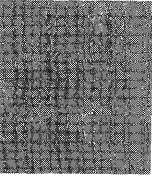
Systems Reference Library

**IBM System/360 Operating System
Conversational Remote Job Entry
Terminal User's Guide**

Program Number 360S-RC-551

The Conversational Remote Job Entry (CRJE) Terminal User's Guide serves primarily as a guide for the inexperienced CRJE user and as a reference for the experienced CRJE user.

The Guide describes the functions of CRJE and the terminal commands that enable the user to perform these functions. The commands and sub-commands are described in detail, and examples illustrate how they may be used. Terminal messages are documented with explanations and corresponding system and user responses.



Second Edition (January 1972)

This publication corresponds to Release 21.

Changes are periodically made to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM System/360 SRL Newsletter, Order No. GN20-0360, for the editions that are applicable and current.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Programming Publications, 1271 Avenue of the Americas, New York, New York 10020. Comments become the property of IBM.

© Copyright International Business Machines Corporation, 1970

The Conversational Remote Job Entry (CRJE) Terminal User's Guide is divided into three major sections: "Using the System," "Command Descriptions," and "Terminal Messages."

The first section, "Using the System," contains tutorial information designed to help the user who is not familiar with the CRJE system. The terminal operations explained in this section are described in detail in the first three appendixes of the book: "Appendix A: 2741 Terminal Operation," "Appendix B: 2740 Terminal Operation," and "Appendix C: 1050 Terminal Operation." Also helpful to PL/I and FORTRAN users are "Appendix D: PL/I Syntax Checker Messages" and "Appendix E: FORTRAN Syntax Checker Messages." These two appendixes contain the messages and their explanation that are sent in response to syntax-checking of PL/I and FORTRAN programs.

The second section, "Command Descriptions," contains reference material consisting of detailed descriptions of the terminal commands and subcommands. These commands and subcommands, along with their functions, are summarized--by subject--in "Appendix G: Terminal Commands and Functions." The formats of the commands are listed alphabetically in "Appendix H: Terminal Command Formats." This appendix briefly summarizes the command formats and is particularly useful to the user who wants a quick review while operating at a terminal.

The last section, "Terminal Messages," contains the CRJE-generated messages--by subject and by number--and includes the message explanations, along with their corresponding system and user responses.

In addition, "Appendix F: Installation Variables," specifies the functions, restrictions, and assignments that are determined by the central installation when the system is generated.

Other publications may be used in conjunction with this guide for further understanding of the following subjects:

1. An overview of the CRJE system--what it provides the system manager, system programmer, terminal user, and central operator--IBM System/360 Operating System: Conversational Remote Job Entry Concepts and Facilities, GC30-2012.
2. The CRJE operating environment--how the system is generated, expanded and maintained--IBM System/360 Operating System: Conversational Remote Job Entry System Programmer's Guide, GC30-2016.
3. The facilities provided for the central operator of CRJE and an explanation of CRJE operation for the central operator--a section of IBM System/360 Operating System: Operator's Reference, GC28-6691.
4. The messages generated by CRJE to be sent to the central operator--IBM System/360 Operating System: Messages and Codes, CG28-6631.
5. An overview of the Operating System (OS)--its concepts and the facilities it provides the programmer--IBM System/360 Operating System: Concepts and Facilities, GC28-6535.
6. An explanation of the control statements that are required for submitting CRJE jobs--IBM System/360 Operating System: Job Control Language Reference, GC28-6704.
7. A detailed explanation of the OS data management facilities--IBM System/360 Operating System: Supervisor and Data Management Services, GC28-6646.

CONTENTS

Introduction	9
Using the System	11
Gaining Access to the System	11
Initiating a Session	11
Ready the Terminal	11
The LOGON Sequence	12
Terminating a Session	13
LOGOFF	14
Another LOGON	14
Abnormal Session Termination	14
Creating, Editing, and Deleting CRJE Data Sets	15
Line Numbering	15
Data Set Attributes	15
Operational Modes	16
Creating a Data Set	18
Making Corrections	19
Listing	19
Saving the Data Set	20
Ending Edit Mode	20
Editing a CRJE Data Set	21
Inserting Lines	21
Deleting Lines	21
Moving Lines	22
Renumbering Lines	22
Saving the Updated Data Set	22
Deleting a CRJE Data Set	23
Syntax Checking PL/I and FORTRAN Statements	23
PL/I Syntax Checker	23
FORTRAN Syntax Checker	24
Automatic Scan	24
Correcting Errors	24
PL/I Statements	24
FORTRAN Statements	25
Scan Subcommand	26
Job Processing	28
Submitting Jobs	28
Identifying Jobs	28
Specifying Input	28
Specifying Output	29
Submitting the Active Data Set	29
Submitting a Data Set from a User's Library	30
Submitting a Data Set List (DLIST)	30
Obtaining Output	32
SYSOUT Output	32
OS Data Sets	34
Canceling Jobs	35
Obtaining Status Information	36
Job Status	36
Data Set Status	36
Sending Messages	37
Central Operator	37
Terminal User	37
Using a Command List (CLIST)	38
Creating a CLIST	38
Executing a CLIST	38
Correcting a CLIST	39
Interrupting Output During CLIST Execution	39
Command Descriptions	41
Command Format	41
Delimiters	41

Continuation	42
Command Notation	42
Format Illustration Conventions	42
Terminology	43
Session Management Commands	44
LOGON	44
LOGOFF	46
DATA MANAGEMENT COMMANDS	46
EDIT	46
Creating a CRJE Data Set	50
Editing a CRJE Data Set	50
Editing an OS Data Set	51
Editing a Data Set After Abnormal Session Termination	52
EDIT Subcommands	52
DELETE	63
Job Processing Commands	64
SUBMIT	64
CANCEL	65
OUTPUT	66
CONTINUE	67
Status Information Commands	68
STATUS	68
LISTDS	69
LISTLIB	70
Message Commands	71
SEND	72
LISTBC	73
Special Commands	73
TABSET	73
EXEC	76
TERMINAL MESSAGES	79
Subject Listing	79
LOGON/LOGOFF	79
COMMAND SYNTAX ERRORS	79
General	79
EDIT	80
EXEC	80
TABSET	80
Library and OS Data Set	80
Active Area	80
SAVE	81
Job Processing (Submission and Output)	81
Status Information	81
SYNTAX CHECKER ERRORS (For statement errors, see Appendixes)	82
User and System Messages	82
System Information	82
Terminal Errors	82
Numerical Listing	82
APPENDIX A: 2741 TERMINAL OPERATION	115
TURNING ON THE TERMINAL	115
END-OF-LINE SEQUENCE	115
ENTERING A LINE	115
CANCELLING A LINE	115
CORRECTING A LINE	115
INTERRUPTING OUTPUT	116
With Interrupt Feature	116
Without Interrupt Feature	116
ERROR CONDITIONS	116
ON-LINE TERMINAL TEST	117
APPENDIX B: 2740 Terminal Operation	119
TURNING ON THE TERMINAL	119
END-OF-LINE SEQUENCE	119
With Automatic EOB	119
Without Automatic EOB	119
ENTERING A LINE	119

CANCELLING A LINE120
CORRECTING A LINE120
INTERRUPTING OUTPUT120
ERROR CONDITIONS120
ON-LINE TERMINAL TEST121
APPENDIX C: 1050 Terminal Operation123
TURNING ON THE TERMINAL123
END-OF-LINE SEQUENCE123
With Automatic EOB123
Without Automatic EOB124
ENTERING A LINE124
With a Request Key124
Without a Request Key125
Cancelling a Line125
CORRECTING A LINE125
INTERRUPTING OUTPUT125
With a Request Key126
Without a Request Key126
ERROR CONDITIONS126
TIME-OUT SUPPRESSION FEATURE127
1050 With Time-Out Suppression127
1050 Without Time-Out Suppression127
ON-LINE TERMINAL TEST127
APPENDIX D: PL/I SYNTAX CHECKER MESSAGES129
Message Format129
User Action129
Errors Detected129
Thorough Checking129
Restricted Checking130
Error Messages130
APPENDIX E: FORTRAN SYNTAX CHECKER MESSAGES141
MESSAGE FORMAT141
USER ACTION141
ERROR MESSAGES141
APPENDIX F: INSTALLATION VARIABLES151
ADDITIONAL COMMANDS AND SUBCOMMANDS151
COMMAND ALIASES151
EXIT ROUTINES151
SYNTAX CHECKERS151
NUMBER OF LINES PER SYNTAX SCAN151
USERID/PASSWORD151
CRJE SYSOUT CLASS152
NUMBER OF LINES PER OUTPUT GROUP152
MAXIMUM NUMBER OF JOBS152
MAXIMUM NUMBER OF MESSAGES152
ROUTING CODES FOR MULTIPLE CONSOLES152
ON-LINE TERMINAL TEST152
APPENDIX G: TERMINAL COMMANDS AND FUNCTIONS153
SESSION MANAGEMENT COMMANDS153
DATA MANAGEMENT COMMANDS153
General153
EDIT Subcommands154
JOB PROCESSING COMMANDS154
STATUS INFORMATION COMMANDS155
MESSAGE COMMANDS155
TABSET COMMAND155
EXEC COMMAND155
APPENDIX H: TERMINAL COMMAND FORMATS157
COMMANDS157
EDIT SUBCOMMANDS158
APPENDIX I: GLOSSARY159

FIGURES

Figure 1. CRJE Modes of Operation 18
Figure 2. CRJE Data Activity. 35
Figure 3. IBM 2741 Communications Terminal Keyboard117
Figure 4. IBM Communications Terminal Model 1 Keyboard with Record
Checking Feature121
Figure 5. IBM 1052 Printer-Keybaord with Panel Switches (x =
Proper Switch Setting)124

TABLES

Table 1. CRJE SYSOUT output for job EXAMPLE 33

Through Conversational Remote Job Entry (CRJE), users at remote terminals can prepare and enter jobs for batch processing under the IBM System/360 Operating System. The remote terminals are connected to an IBM System/360 central processing unit (CPU) via communication lines.

A user at one of these terminals accomplishes his work through the use of the terminal command language. He requests system functions by entering commands that are interpreted and processed at the central system. When a request has been serviced, the user is notified (by either a message or a one-character indicator), and he can proceed with another command.

The period of time during which a terminal user engages in conversation with the CRJE system is called his session. A session begins when a user logs on the system by entering a command that identifies him as an authorized user. There is another command with which he logs off, thereby terminating his session. While he is logged on, a user is considered active.

On command from a terminal user and under the control of CRJE, jobs are entered into the OS job stream for batch processing. Execution of remotely submitted jobs proceeds under the supervision of the OS job management routines, and the data sets comprising the job output are created using OS data management facilities.

Job input consists of programs and data in data sets that are conversationally created and maintained through the updating facilities of CRJE. Creating and maintaining data sets is accomplished through the use of a temporary work area, the active area. There is a command - EDIT - that allows the terminal user to create a temporary data set called an active data set. Initially, the user enters lines of program source statements, data, and job control language (JCL) that are collected in the active area. This active data set can optionally be retained, on command, in the user's permanent library.

Data sets in a user library can be updated by creating an active data set that is a copy of the permanent version. Changes to this data set are made using a group of commands (the EDIT subcommands) that allow inserting, replacing, deleting, and modifying single lines or groups of lines. The user can save the updated version in his library, either as an additional data set or as a replacement of the old version.

Before submitting a job for processing, the terminal user may request that FORTRAN or PL/I source statements be checked for correct syntax. He will receive error diagnostics immediately and can take corrective action; this eliminates much compilation cost and waiting for a complete job turnaround.

The listing facility of CRJE allows the user to examine all or portions of job output and of data sets in user libraries. Data sets in user libraries and certain cataloged data sets that are created through OS data management facilities can be listed if they are copied into the active area. Job output directed to a special system output class for CRJE can be listed without first being copied into the active area.

Information about the status of jobs that he has submitted and about data sets in his library is available for the terminal user. In addition, he can send messages to the central operator and to other CRJE users.

GAINING ACCESS TO THE SYSTEM

In order to use the facilities of CRJE, you must have a userid and a password; these are assigned to you by your system programmer. The userid is your individual code, which identifies you to the system when you log on and which is used by anyone sending you a message or gaining access to your user library. A password is associated with your userid and must also be specified when you log on to identify you as an authorized system user.

As a CRJE user, you also need to know how to operate the terminal you will be using - how to turn it on, how to enter a line of input, how to indicate the end of a line of input, and how to interrupt the transmission of output to your terminal. There are standard procedures for canceling a line and for correcting errors in a line that has not been terminated with an end-of-line indicator. Operating procedures vary according to the type of terminal you are using. Your terminal may be an IBM 2741 Communications Terminal, an IBM 2740 Communications Terminal, or an IBM 1050 Data Communications System. Instructions for operating the 2741 are in "Appendix A: 2741 Terminal Operation"; the 2740, in "Appendix B: 2740 Terminal Operation"; and the 1050, in "Appendix C: 1050 Terminal Operation." Be sure that you read the appropriate appendix before you attempt to log on at the terminal.

Whenever the system is ready for you to enter a command, you will be prompted with an underscore, followed by a backspace and the unlocking of your keyboard. You may enter commands in either uppercase or lowercase characters. Throughout this publication, examples show user entries in lower case and system responses in upper case.

In general, commands consist of an operation field, which contains the command verb, and an operand field, which contains the operands that further qualify the operation. One or more blanks must be used to separate the operation field from the operand field. Unless specifically noted, delimiters separating operands in the operand field may be one or more blanks, a single comma, or a combination of a single comma and one or more blanks.

INITIATING A SESSION

When you are ready to use CRJE, you must ready your terminal for use and must enter the LOGON command to initiate your session.

Ready the Terminal

Instructions for turning on the terminal are in the appendix that gives operating procedures for the type of terminal you are using.

If your terminal is permanently connected to the system, turning on the terminal makes the communication line connecting it to the system ready for use.

If your terminal has a telephone-like device called a modulator-demodulator, or modem, you must dial the number of the system to establish the communication line connection after you turn the terminal on. To dial the system number:

1. Press the TALK button and lift the receiver.
2. Dial the number of the central computer. (Your system programmer will give you the number.)
3. Press the DATA button when you hear a continuous high-pitched tone, and place the receiver back in its cradle. The terminal is now connected to the system.

The LOGON Sequence

After you have turned on the terminal and, if necessary, established the connection with the system, you must enter a null line to indicate that you are ready to type in the LOGON command. A null line is generated by entering the end-of-line sequence for your terminal type without entering any characters. See the appropriate appendix for details.

After you enter a null line, the system will respond with an underscore to prompt you for your LOGON command.

Your userid and password must be specified as operands of the LOGON command. Your installation may also require certain accounting information on the command. If this is the case, your system programmer will tell you what information to supply.

Your LOGON command typed after the underscore prompt will look like this:

```
logon brown/john
```

Note that the userid, BROWN, and the password, JOHN, are separated only by a slash (/). If accounting information is required, the command will look like this:

```
logon brown/john acct(457360)
```

ACCT is a keyword that is recognized by the system as an indication that accounting information, enclosed in parentheses, follows.

If either the userid or the password is entered incorrectly, or if the system does not recognize the pair as a valid combination, you will receive a message requesting the correct information. If the userid is invalid, missing, or currently in use, the message will be ENTER USERID, after which you can enter just your userid. If the password is invalid or missing, the system will print ENTER PASSWORD ■■■■■■ and will backspace to the beginning of the overprinted area. You can then key in your password so that no one looking at the listing can see what your password is. If your response to either of these prompting messages is unacceptable, the LOGON command is rejected and you must start the entire LOGON procedure again.

When your LOGON command is accepted, the system will respond with a message acknowledging you as an active user by printing the address of the communication line connecting your terminal to the system, the date, and the time.

After the LOGON acknowledgment message, you will receive any messages that were generated for you by the system or sent to you by the central operator or another user while you were inactive. A message sent to you by another user or the central operator is preceded by the phrase CRJE MSG FR, followed either by the word CENTRAL for the central operator or by the userid of the user sending you the message. Throughout your session, messages are generally delivered to you after a command has been processed so that listing and input operations are not interrupted. (There may be some instances when a message is sent to you before a command is completely processed; e.g., if you enter a LISTBC command, the broadcast messages will be listed after any pending system messages have been sent.)

If you anticipate trouble understanding messages generated by CRJE, you can specify an additional operand, MSGID, on the LOGON command, to request that message codes be sent with messages. Then you will have no trouble looking up the explanation and user action in the second part of the CRJE Messages section of this publication, where the messages are listed in numerical order according to message codes.

If you do not specify the MSGID operand, message codes will not be sent with CRJE messages. To find the message code of a message that you wish to look up, use the first part of the CRJE Messages section, where message texts and codes are listed according to subject.

Another optional operand of the LOGON command, BC, specifies whether or not broadcast messages are to be sent following the LOGON command. Unless you specifically request that they not be sent, you will receive the current broadcast messages, which the central operator generates for all users. These messages are preceded by the letters BRD and a message identifier, for example,

```
BRD0010 CLOSEDOWN WILL BE AT FIVE
```

```
BRD0020 STARTUP TOMORROW AT NINE
```

If you want a copy of the broadcast messages at any other time during your session, you can request it by entering the following command:

```
listbc
```

If your terminal is not equipped with the special Interrupt feature, all output - messages, job output, and data set listings - is sent to you "n" lines at a time, where "n" is determined by the central installation. Therefore, when there is a break in output being printed but no underscore prompt for the next command, the system is probably waiting for an indication from you to continue or to interrupt the transmission of output to your terminal. Between each group of lines, you can take appropriate action to receive the next group of lines, to receive the remaining output without interruption, or to stop receiving output. See the appendix on the operation of your terminal type for instructions on interrupting output.

If you interrupt output while receiving messages, any undelivered messages are deleted, and any pending request for the broadcast messages is canceled.

TERMINATING A SESSION

You can end your terminal session by entering a LOGOFF command or by allowing another user to enter a LOGON command. Do not just turn off the terminal because, depending on the terminal type and the

communication line connection, the system will either detect a line error or assume you are still active.

LOGOFF

If you are through using CRJE for the time being and you are not working with an active data set, enter a LOGOFF command, which has no operands. If you have been working with an active data set, be sure you have entered the END subcommand before trying to log off. (See the following section on editing and creating data sets for details).

logoff

LOGOFF 10.10 SESSION TIME 0030 MIN.

Notice that when you log off, the system sends a message giving you the time, according to the system clock, that your LOGOFF was processed, and the total number of minutes of the session just ended.

After logging off, you may turn off the terminal unless you are using a 1050 terminal with a permanent (leased) line connection. (Turning power off on a 1050 on a leased line causes a line error that results in deactivation of the line. The central operator will have to reactivate the line with a central command before the terminal can be used again.) If the line connection was established by dialing, the system automatically disconnects the line.

Another LOGON

If you have finished using CRJE and there is another user ready to start a session, you may allow him to enter his LOGON command without your entering LOGOFF. You are automatically logged off the system at the same time that he is logged on. This is especially useful if the terminal is connected over a switched communication line (i.e., one which requires dialing), since the line is not disconnected.

ABNORMAL SESSION TERMINATION

You will be automatically logged off the system if there is a transmission error on the communication line connecting your terminal to the central system, or if there is a normal or abnormal closedown of the central system. If you are working with an active data set at the time, CRJE will attempt to save it with the dsname ACTIVE in your user library.

After your session has been terminated, you may begin another session by entering another LOGON command. If operating on a switched line, you must reestablish the connection by dialing. If your session was terminated because of a central closedown, you cannot begin a new session until the central system is operating again.

CREATING, EDITING, AND DELETING CRJE DATA SETS

The data you enter from your terminal is collected in the active area, a work area (on disk) that you share with other users. The lines of data that you enter comprise your active data set. The active area is the intermediate step between you and your user library, where your data sets can be stored permanently. The data sets in your user library are called CRJE data sets. You can list and update CRJE data sets in your library by having them copied into the active area, where changes can be made. The updated version can then be stored in your library as a new data set or as a replacement of the original version. You use the EDIT command to create an active data set or to copy a data set into the active area, where it becomes your active data set.

LINE NUMBERING

Each record in the active data set, and consequently in a CRJE data set, corresponds to a line entered from the terminal. A line number is associated with each record, or line, so that you can refer to specific lines when requesting a listing or a modification. The maximum line number that can be associated with a line is 99,999,999. When a listing or a modification is requested, the system uses the first eight positions of the record for the actual line number, then skips two more spaces to separate the line number from the actual line of input. Therefore, the first ten spaces of each record are reserved by the system for line numbers.

Note: This format of a record applies only while the record is in the active area. Another format may be used when the record is being saved in your user library. See "Saving the Updated Data Set" in this section.

In addition to maintaining line numbers internally, CRJE will maintain the line numbers in the last eight positions of each line entered unless you request otherwise. (See the NONUM operand in the EDIT command description). Each input line you enter is padded with blanks or truncated to either 72 or 80 characters, depending upon whether or not line numbers are kept in positions 73-80.

Note: The user must include a line number in each record to be saved in an 80-character record library.

DATA SET ATTRIBUTES

Attributes are assigned to a data set when it is created to indicate certain characteristics of the data set. The attributes originally assigned remain associated with the data set as long as it is in your user library, unless you specify a change.

There are three types of attributes associated with a data set: the content attribute, the sequence attribute, and the line increment attribute.

The content attribute indicates what type of records are contained in the data set. Possible content attributes are:

PL1: The records are PL/I source language statements that can be syntax checked if a PL/I Syntax Checker is provided in the system.

- FORT:** The records are FORTRAN source language statements that can be syntax checked if a FORTRAN Syntax Checker is provided in the system. The language level - E, G, or H - is also part of the attribute, e.g., FORTE, FORTG, FORTH.
- DATA:** The records consist of data in which alphabetic characters are maintained in upper case. A data set containing source language statements, Job Control Language (JCL) statements, or possibly data input to a program would have the DATA attribute. If you do not specify a content attribute when you create a data set, the system automatically assigns the DATA attribute.
- TEXT:** The records consist of data in which alphabetic characters can be in either upper case or lower case. CRJE does not translate lowercase alphabetic characters to upper case in any of the lines entered for a data set with the TEXT attribute.
- DSLIS:** Each record contains the name of one CRJE data set or an asterisk (*) to indicate the active data set. Nesting of DSLIS data sets is not allowed - that is, a record in a DSLIS cannot be the name of a data set that has the DSLIS attribute. The use of a DSLIS data set is described later, in the section "Job Processing."
- CLIS:** Each record is a CRJE terminal command or subcommand. A command in a CLIS data set cannot be continued on a second line. The uses of a CLIS, or command list, are described in the section "Using a Command List."

The sequence attribute indicates whether line numbers are kept in positions 73-80 of each record. These line numbers are maintained in addition to line numbers automatically appended to each record. The two possible sequence attributes are:

- SEQ:** The records contain line numbers in positions 73-80. Your input to a data set with this attribute is limited to 72 characters. If you do not specify a sequence attribute when you create a data set, the system automatically assigns the SEQ attribute.
- NOSEQ:** The records do not contain line numbers in positions 73-80. All 80 positions are therefore available for your input.

Note: The SEQ and NOSEQ attributes are assigned with the NUM and NONUM operands, respectively, on the EDIT command.

The line increment attribute is the difference between line numbers in the data set. The increment is used when the data set is created and whenever lines are added to the data set. The system uses a line increment of 10. The only way you can change the increment is by renumbering the data set.

OPERATIONAL MODES

While using CRJE, you operate in one of three modes: command mode, edit mode, or input mode. The type of input being entered determines the mode in which you are operating.

From the time you log on the system and whenever you do not have an active data set, you are in command mode. Since the EDIT command results in the creation of an active data set, entering EDIT takes you

out of command mode. No more commands are accepted until you return to command mode by entering the END subcommand, which deletes the active data set.

Following the entry of an EDIT command, you are in either edit mode or input mode until you enter END. In edit mode, you enter EDIT subcommands, which are used to list and manipulate lines in the active data set. In input mode, you enter lines that are to be records of the active data set. A null line can be used to switch from input mode to edit mode and vice versa. Figure 1 summarizes the three modes of operation in CRJE.

Input mode: Input mode is entered in one of three ways:

1. When you enter an EDIT command to create a new data set. The system will begin prompting you for input by printing the line number of the line to be entered. Prompting begins with line number 10 and continues in increments of 10 until you respond with a null line, which puts you in edit mode.
2. When you enter an INPUT subcommand, which is used to insert and replace lines in the active data set. Unless you request not to be prompted, the system prompts you for input according to your specifications on the subcommand. When you have finished entering lines, a null line returns you to edit mode.
3. When you enter a null line in edit mode (i.e., a null line in response to an underscore prompt for an EDIT subcommand). The system will begin prompting you for input, starting at the end of the data set and continuing in increments of whatever line increment attribute is assigned to the data set. Again input mode is terminated by entering a null line in response to a line number prompt.

Edit Mode: In edit mode, you enter an EDIT subcommand in response to an underscore prompt. Subcommands can be abbreviated whereas commands entered in command mode cannot. Edit mode is entered in the following ways:

1. When you enter an EDIT command to obtain a copy of an existing data set for listing or updating. You will receive an underscore prompt for an EDIT subcommand when the data set has been copied into the active area.
2. When you enter a null line in input mode. The system will then prompt you for a subcommand by printing an underscore.

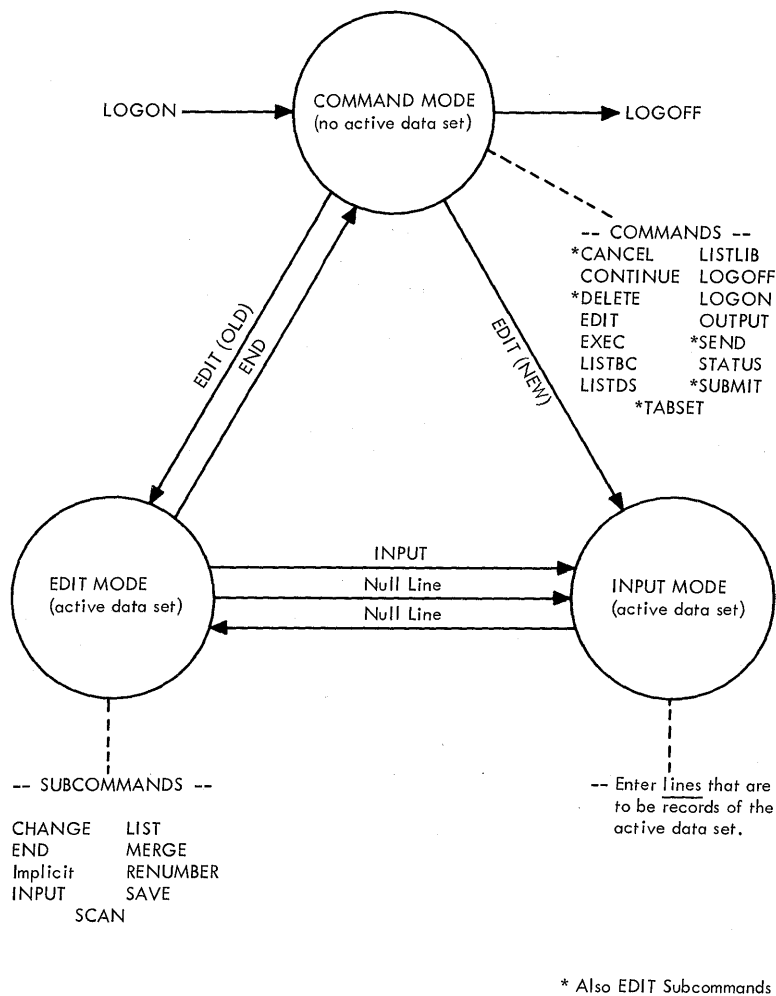


Figure 1. CRJE Modes of Operation

CREATING A DATA SET

To create a data set, enter the EDIT command, specifying the name for the data set and the NEW keyword operand to indicate it is a new data set.

```
edit mydata,new
```

In this example, MYDATA is the data set name, which must contain 1-8 characters, the first of which must be alphabetic. The content attribute is DATA by default, so all lowercase alphabetic characters entered will be translated to upper case. Line numbers will be maintained in positions 73-80 of each line, since the SEQ sequence attribute is assigned by default.

You enter input mode and the system begins prompting with line number 10, using an increment of 10 between records. You type in a line following each line number prompt, terminating each line with the end-of-line sequence.

10 line number one
20 line number two
30 line number three
40 line numbr four
50 line number six
60

Making Corrections

The system has prompted you for line number 60 when you notice you have made errors in line 40 and line 50. You must enter a null line to enter edit mode so that you can enter subcommands to correct the errors.

The Implicit and CHANGE subcommands are particularly useful in correcting errors of this type.

To correct the spelling error in line 40, for example, you may use the CHANGE subcommand, abbreviated C, to replace the characters in error without having to reenter the entire line.

```
c 40 /br/ber/
```

Br is replaced by ber. (Line 40 is expanded to accommodate the additional character.) The characters in error (br) and the replacement characters (ber) are delimited by a single character - in this case, a slash (/). The delimiter is the first character, other than a blank or a comma, after the line number.

The Implicit subcommand can be used to enter an entire line without being in input mode. You simply enter the line number, a single blank or comma as a delimiter, and the text of the line. To correct the number in line 50, you may replace the line by entering

```
50,line number five
```

or you may insert a line by entering

```
45,line number five.
```

The system will print another underscore to prompt you for your next subcommand. If you would like to continue entering lines in the data set, you can enter a null line. Prompting will resume with line number 60. If you inserted line 45 to correct the error above, you might continue as follows:

```
60 line number seven  
70 line number eight  
80 line number nine  
90 line number ten  
100
```

Assuming ten lines complete the data set, you enter a null line after line number 100, receiving an underscore prompt in response.

Listing

You can list the lines in your active data set by entering the LIST subcommand, abbreviated L. You may specify a single line or the first and last line numbers of a range of lines to be listed; or you may list the entire data set by not specifying any operands.

Although the lines in this data set contain line numbers in positions 73-80, the line numbers are listed preceding the text. If you do not want line numbers listed at all, you can specify the NONUM operand, which is explained in the LIST subcommand description in the reference section of this manual.

Listing the entire data set, you see that your corrections were made. Also note that lowercase alphabetic characters have been translated to upper case.

1

```
10 LINE NUMBER ONE
20 LINE NUMBER TWO
30 LINE NUMBER THREE
40 LINE NUMBER FOUR
45 LINE NUMBER FIVE
50 LINE NUMBER SIX
60 LINE NUMBER SEVEN
70 LINE NUMBER EIGHT
80 LINE NUMBER NINE
90 LINE NUMBER TEN
```

Saving the Data Set

Use the SAVE subcommand if you wish the active data set to become a permanent CRJE data set in your user library. SAVE can be abbreviated S and does not have any required operands.

S

The data set named MYDATA is now stored in your library. Another user who knows the name of the data set and your userid has access to MYDATA, although he cannot delete or replace the copy of MYDATA that is in your library. If you wish more protection, you can assign a key to the data set so that anyone wanting access to it must know the key. The operand used to assign a protection key is discussed with the description of the SAVE subcommand.

Note: If you have an 80-character record user library instead of an 88-character one, which is assumed in this example, data located in positions 73-80 of each record will be replaced with the line number when the record is being saved. The system sends message IHK410 DATA LOST IN TRUNCATION when this occurs. If you do not need this data, enter the END subcommand. Otherwise, reenter the data on as many lines as needed, making sure that none of it is located in positions 73-80.

Ending Edit Mode

When you are through using the current active data set and have saved it if necessary, you must enter the END subcommand to delete the active data set and return to command mode.

end

The next line entered must be a command. edit subcommands will not be accepted until another EDIT command is entered to provide another active data set.

EDITING A CRJE DATA SET

At a later time, you may wish to modify the data set MYDATA, which you saved in your user library. The EDIT command is used to initiate editing operations on the data set. The only operand necessary is the data set name, since attributes are already assigned and the system assumes by default that the data set exists.

```
edit mydata
```

When this command is entered, mydata is copied from your library into the active area, where it becomes your active data set. Notice that, following an EDIT command for an old data set, you enter edit mode, receiving an underscore prompt rather than a line number prompt. You are now ready to enter EDIT subcommands.

Inserting Lines

Suppose you wish to add lines after line 20 in the data set. With the INPUT subcommand, you can specify the line number of the first line you want to enter and the increment to be used between succeeding lines. The increment specified is used only as long as you are in input mode following this INPUT subcommand. It does not become the line increment attribute. Following INPUT, you are in input mode and will receive line number prompts unless you specify otherwise (see the description of the INPUT subcommand for details).

```
i 22,2  
22 line number two-a  
24 line number two-b  
26 line number two-c  
28 line number two-d  
30
```

You do not wish to replace the existing line 30, so you enter a null line when prompted for line 30. To guard against accidentally replacing an existing line, you may specify insertion only by entering I as an additional operand. If I had been specified in the preceding example the system would have sent a message instead of the prompt for line number 30.

The INPUT subcommand can also be used for replacement of existing lines only. If you specify a starting line number and an R, the system will prompt you with line numbers of existing lines, starting with the number you specify.

Deleting Lines

The DELETE subcommand can be used to delete a line or a range of lines in the active data set. To delete line 28, for example, you may enter

```
d 28
```

The Implicit subcommand can also be used to delete a single line. You simply enter the line number of the line to be deleted. The following accomplishes the same results as the example above:

Moving Lines

You may move lines in the active data set by specifying, with the MERGE subcommand, the lines to be moved and the line after which they are to be inserted. The lines are copied from the original to the new location, and the original lines are not deleted. If no new location is specified, the lines will be copied following the last line of the data set. For example, to copy lines 70 through 90 onto the end of the data set, you would enter:

```
m * 70,90
```

The asterisk (*) indicates that you are merging lines within the active data set. The MERGE subcommand may also be used to merge lines from a data set in a user library into the active area.

When lines 70-90 are added at the end of the data set, they are numbered according to the line increment attribute of the data set. In this case, they will be numbered 100, 110, and 120.

Renumbering Lines

Now that your updates have been made, you may wish to renumber all the lines in the active data set to allow for future insertions. The RENUMBER subcommand with no operands causes all the lines to be renumbered starting with line number 10 and using an increment of 10.

```
ren
```

You may specify another starting line number and increment in the operand field. The increment specified becomes the line increment attribute of the data set.

Saving the Updated Data Set

You may now wish to list the updated data set, which is still in the active area, save it in your library, and terminate editing. When saved in your library, it will replace the original version. If you wish to keep the original data set along with the updated version, you can give the updated data set a different name via the dsname operand of the SAVE subcommand.

```
l
```

```
10 LINE NUMBER ONE
20 LINE NUMBER TWO
30 LINE NUMBER TWO-A
40 LINE NUMBER TWO-B
50 LINE NUMBER TWO-C
60 LINE NUMBER THREE
70 LINE NUMBER FOUR
80 LINE NUMBER FIVE
90 LINE NUMBER SIX
100 LINE NUMBER SEVEN
```

110 LINE NUMBER EIGHT
120 LINE NUMBER NINE
130 LINE NUMBER TEN
140 LINE NUMBER EIGHT
150 LINE NUMBER NINE
160 LINE NUMBER TEN

s

end

The next line you enter must be a command.

DELETING A CRJE DATA SET

To remove a data set from your user library, use the DELETE command. DELETE must be entered in command mode and cannot be abbreviated as can the DELETE subcommand, which is used to delete lines in the active data set. The data set name is the required operand.

delete mydata

SYNTAX CHECKING PL/I AND FORTRAN STATEMENTS

If the appropriate syntax checker is included in your installation's system, you can request that your PL/I or FORTRAN statements be scanned for correct syntax. Diagnostics are performed on a single statement basis only, and errors requiring cross-checking between different statements are not detected.

Syntax checking is performed on lines in the active data set or lines being entered in input mode. The automatic scan facility provides automatic checking of source statements as they are entered in input mode, while the SCAN subcommand is used to request checking of lines already in the active data set. One statement at a time is passed to the syntax checker. However, if one line contains more than one PL/I statement, the entire line, rather than just one statement, is passed at once. PL/I and FORTRAN statements can be continued on succeeding lines, and the entire statement can be checked as long as an installation-imposed maximum number of continuation lines is not exceeded.

When an error is detected, a diagnostic message is produced. A syntax checker attempts no corrective action. Multiple errors in a statement can be diagnosed only if the continued scan through the statement does not depend on any corrective action.

PL/I SYNTAX CHECKER

If the PL/I Syntax Checker is included in the system, it provides either thorough checking or restricted checking for the latest version of the F-level PL/I language that is available on the system. The syntax checker that provides thorough checking will detect almost all errors that are dependent on information present in that single PL/I statement. When the level of checking is restricted, the syntax checker will detect a subset of the thorough checking errors. Compile-time statements are ignored by the PL/I Syntax Checker.

Error messages produced by the PL/I Syntax Checker are listed in "Appendix D: PL/I Syntax Checker Messages."

FORTRAN SYNTAX CHECKER

A FORTRAN Syntax Checker included in the system provides checking for the basic FORTRAN IV language (level E), the full FORTRAN IV language (levels G and H), or both the basic and full language. In scanning a statement, elements may be detected that are erroneous for all three language levels. In such a case, the same diagnostic is produced regardless of the language level specified. There are, however, errors distinctive to each language level. These errors relate to features not supported at a particular level, such as the use of full FORTRAN IV language features not allowed in FORTRAN E.

Error messages produced by the FORTRAN Syntax Checker are listed in "Appendix E: FORTRAN Syntax Checker Messages."

AUTOMATIC SCAN

The automatic scan facility can be turned on by the SCAN operand of the EDIT command or the ON operand of the SCAN subcommand. While the automatic scan is on, all lines entered in input mode are passed to the syntax checker. The automatic scan is turned off by the END subcommand or by the SCAN subcommand with the OFF operand.

Note: Lines entered with the Implicit subcommand are not syntax checked.

Correcting Errors

If a syntax error is detected, you will receive a diagnostic message indicating the line number of the input line that contains the error, followed by an underscore prompt for a subcommand to correct the error. You can take one of two actions:

1. Ignore the error by entering a null line to resume input mode.
2. Correct the error by using either the Implicit or CHANGE subcommand. You may make as many corrections as you wish, as long as each line corrected is the line, or within the range of lines, containing the statement just scanned. When the error has been corrected, enter a null line, which causes the lines to be rechecked and input mode resumed.

PL/I Statements

A data set containing PL/I statements to be syntax checked must have the PL1 content attribute, which is assigned with the EDIT command when the data set is created. If you are not using the 60-character set and 2 and 72 as your left and right source margins, see the description of the EDIT command for the parameters you must specify.

When using the automatic scan, you must indicate continuation lines so an entire statement can be collected before being passed to the syntax checker. To indicate that the next line is a continuation line, immediately precede the end-of-line sequence for the current line with a hyphen (-). The hyphen is not saved in the active data set.

The following example illustrates the creation of a data set containing PL/I statements that are scanned automatically as they are entered.

```

edit pliprogram new pl1 s
  10 procedure options (main);
  10 LABEL MISSING FROM PROCEDURE STATEMENT
c 10 /p/std dev:p/
  20 /*this procedure computes the standard deviation for a given-
  30 frequency distribution, represented by a set of records each-
  40 containing a value and a frequency*/
  50 declare value(100) fixed(5,2), freq(100) fixed(3),-
  60      (mean,standev,devn) fixed(5,2) sqdev fixed (10,4),-
  70      (sumsqdev fixed(14,4), n fixed(4), sum fixed(9,2))-
  80      init(0), (i,k) fixed(3) inti(1);
  60 INVALID SQDEV ATTRIBUTE SPECIFIED FOR MEAN
c 60/2)s/2),s/
  90 on endfile(sysin) do; k = i; goto compmn; end;
  90 ILLEGAL STATEMENT FOLLOWS ON
c 90/do/begin/
  100 next: get list (value(i), freq(i);
  100 UNMATCHED PARENTHESES , ) MISSING
c 100/;/)/;/
  110 sum = sum + value(i)*freq(i); n=n+freq(i); i=i+1; goto next;
  120 compmn:mean = sum/n;
  130 do i=1 to k; devn = value(i) - mean; sqdev = devn * * 2;
  130 INVALID PREFIX OPERATOR * PRECEDING 2
c 130 /* */**/
  140 sumsqdev = sumsqdev + sqdev * freq(i); end;
  150 standev = sqrt(sumsqdev/n);
  160 put edit ('frequency table', (value(i),freq(i), do i=1 to k)-
  170      , 'standard deviation',standev) -
  180      (page,line,a,skip(1),x(2),(k) (skip,x(5),p'+-99',-
  190      x(4),p'zz9'), skip(2),a,x(5),p'zz9.99');
  160 OPERAND MISSING BEFORE DO
  180 LEFT PARENTHESIS REQUIRED AFTER LINE
  180 MORE THAN ONE SIGN CHAR IN PICTURE BEGNG '+-99'
c 160/),d/) d/
  180 (page, line(5),a,skip(1),x(2),(k)(skip,x(5),p'+99',
  190 end;

```

FORTRAN Statements

A data set containing FORTRAN statements to be syntax checked must have the FORTE, FORTG, or FORTH content attribute to indicate the language level of the syntax checker to be used. The attribute is assigned with the EDIT command when the data set is created.

When using the automatic scan, you must indicate continuation lines so an entire statement can be collected before being passed to the syntax checker. To indicate that the next line is a continuation line, immediately precede the end-of-line sequence for the current line with a hyphen (-), which is not saved in the active data set. Position six of the continuation line must contain a character other than a blank or a zero.

The following example illustrates the creation of a data set containing FORTRAN statements that are scanned automatically as they are entered.

The TABSET command is used to facilitate entering the FORTRAN statements. (TABSET is meaningless when used with the Implicit subcommand. Therefore, if corrections are to be made in the FORTRAN statements, it is advisable to use the CHANGE subcommand.)

Note: Tabs are not used for the statements containing comments. Where statements contain statement numbers, depress the TAB key and backspace three spaces, then enter the statement number followed by a blank and the FORTRAN statement. For continuing a statement, depress the TAB key, backspace once, and enter the continuation character, plus the balance of the FORTRAN statement.

edit ftnprog, new, fortg, s

```

tabset 6
  10 c      this program generates and prints the first twenty rows
  20 c      of a triangle of binomial coefficients
  30 c
  40        dimension koeff ( 20, 20 )
  50        data koeff / 400*1 /
  60 c
  70        do 20 i = 2, 19
  80        do 10 j = 2,
80 UNSIGNED INTEGER EXPECTED
c 80 /2,/2, i/
  90        koeff ( i+1, j ) = koeff ( i, j ) + koeff ( i, j-1 )
 100        10      continue
 110        20 cOntinue
110 CONTIN NAME TOO LONG
c 110 /c0/co/
 120 c
 130        write ( 6, 1000 )
 140        1000 format ( '1 table of the binomial coefficients for n
                        from 0 to 19'-
 150        * /computed using integer arithmetic, without
                        computing any fa-
 160        * ctorials' / ' ' )
150 COMPUT DELIMITER MISSING OR INVALID FORMAT CODE
c 150 a/a/'a
 170        do 30 i = 1, 20
 180        write ( 6, 2000 ) ( koeff ( i, j ), j = 1, i
180 ) REQUIRED FOR IMPLIED DO
c 180* i * i )*
 190        30      continue
 200        2000 format ( ' ',i1,19i6 )
 210        stop
 220        end
 230

save
end

```

SCAN SUBCOMMAND

The SCAN subcommand can be used to check the syntax of statements contained in the active data set. You can request checking of a single statement, a group of statements, or the entire data set.

When a syntax error is detected, you will receive a diagnostic message indicating the line number of the line that contains the error,

and checking continues with the next statement if there are more statements to be scanned. You are not required to take any corrective action or any action to indicate you do not wish to make corrections. When the subcommand has been processed completely, you will receive an underscore prompt for your next subcommand.

If you specify one line number in the operand field of the SCAN subcommand, that line is passed to the syntax checker, along with its continuation lines if a statement does not end on that line. For example, suppose the FORTRAN data set, FTNPROG, had not been scanned automatically and still contained the errors originally entered. If you enter the following SCAN subcommand,

```
sc 140
```

all three lines, 140-160, are passed to the syntax checker, since 150 and 160 are both continuation lines. You will then receive the error message

```
150 COMPUT DELIMITER MISSING OR INVALID FORMAT CODE
```

the underscore prompt for a subcommand is printed because processing is complete since only one statement was to be scanned.

If you wanted to scan all statements from line 140 to the end of the data set, you could enter

```
sc 140, 250 .
```

The second line number can be any number equal to or greater than 220, the last line in the data set. The response to this subcommand will be

```
150 COMPUT DELIMITER MISSING OR INVALID FORMAT CODE
```

```
180 )REQUIRED FOR IMPLIED DO
```

To request that the entire data set be scanned, no operands are needed on the SCAN subcommand:

```
scan
```

You will receive all the error messages, and when scanning is completed, you will be prompted for another subcommand.

```
80 UNSIGNED INTEGER EXPECTED
```

```
110 CONTIN NAME TOO LONG
```

```
150 COMPUT DELIMITER MISSING OR INVALID FORMAT CODE
```

```
180 ) REQUIRED FOR IMPLIED DO
```

You may correct errors as you wish, using any of the EDIT subcommands. There is no automatic rechecking of corrected statements. If you want your corrections checked, you may enter corrective lines in input mode using the automatic scan facility, or you may reissue the SCAN subcommand after making appropriate changes.

JOB PROCESSING

SUBMITTING JOBS

In order for you to submit a job for execution, all the input necessary in the job stream must be contained in a CRJE data set or a combination of CRJE data sets. Jobs can be submitted using the SUBMIT command while you are in either command mode or edit mode. If you are in edit mode, part or all of the input can also be in your active data set.

Since jobs submitted through CRJE are processed in the standard batch environment, the input submitted for processing is the same as input you would submit through a local card reader. The input, which is entered into the OS job stream, must include all the Job Control Language (JCL) statements necessary for the desired processing.

The JCL describes the job to be processed, the input data sets used in the processing, and the output data sets to be created by the job. You may supply your own JCL, or it may be provided for you in a CRJE data set in a user library. This JCL may invoke a procedure from the OS procedure library.

Note 1: CRJE does not support the use of instream procedures under MFT.

Note 2: The user must guard against using JOBFAIL as the name of a remotely submitted job because the operating system assigns that name to all jobs having JCL errors on the JOB statement.

Identifying Jobs

The symbol in the name field of the JOB statement is the jobname, which identifies the job while it is in the system. When you inquire about the status of a job, you identify the job by specifying the jobname. No two jobs in the CRJE system at the same time can have the same jobname. A job is considered to be in the CRJE system from the time it is submitted by a user until either all the terminal output has been returned to the user, the job has been canceled, or the operator has obtained the terminal output, with a central command.

Specifying Input

Any OS data sets - that is, data sets not created through CRJE - may be used in job processing through the data definition facilities of JCL. A CRJE data set or the active data set used as an input data set should be defined by a DD * or DD DATA statement, with a /* at the end of the input. If the DD DATA statement is used, DATA must be the only parameter in the operand field. However, DD DATA statements should not be used in cataloged procedures or instream procedures. (An instream procedure is a set of uncataloged statements in card form that has been placed in the input stream of the card reader; it can be executed any number of times during the job in which it appears.)

Note: CRJE does not support the use of instream procedures under MFT.

Specifying Output

The destination of any output produced by a job is controlled by the Data Definition (DD) statements submitted with the job. There are two kinds of output data sets that you can obtain at your terminal: data sets in the CRJE system output (SYSOUT) class, and named OS data sets that are cataloged, resident on a single volume of a direct-access storage device, and either sequential or partitioned. Volumes containing OS data sets to be obtained at remote terminals must have been mounted at CRJE system start-up.

If you need only one copy of an output data set returned to your terminal, you may specify the CRJE SYSOUT class on the DD statement for that data set. The CRJE SYSOUT class, which is assigned by your installation, contains output to be returned to remote terminals. To obtain output from this SYSOUT class, you simply specify the jobname of the job that produced the output.

If you will need to list an output data set more than once, if you will want to list portions of the data set, or if other users will also want access to the output data set, you should name the data set and specify that it be created on a direct-access volume, be cataloged, and be either sequential or partitioned. The data set can then be obtained, using the EDIT command, by anyone knowing the fully qualified name of the data set. If the first qualifier of the data set name is your userid, you will be able to delete the data set with a CRJE command (DELETE).

The CRJE SYSOUT class is the default if the MSGCLASS parameter of the JCL JOB statement is not specified for system messages. These system messages, also called job management messages, include allocation and deallocation messages, diagnostics of JCL statements, etc. Specifying any MSGCLASS other than the CRJE MSGCLASS may result in the loss of the job. If the job is lost, the user cannot resubmit that job with the same jobname unless the CRJE procedure is restarted. Furthermore, no other user may submit a job with the jobname in question (he will get the message, IHK326 DUPLICATE JOBNAME).

Any output assigned to SYSOUT classes other than the CRJE SYSOUT class is directed to the corresponding devices at the central installation.

Submitting the Active Data Set

You can submit an input stream for processing while you are in edit mode. For example, suppose the FORTRAN program in the data set FTNPROG is in your user library and you wish to compile, link edit, and execute it. You can make it your active data set, insert the appropriate JCL, and submit it directly from the active area. The CRJE SYSOUT class in this example is R.

```

edit ftnprog
i 0, 2
  00 //example job msglevel=1,msgclass=r
  02 // exec fortgclg
  04 //fort.sysin dd *
  06
i
230 /*
240 //go.ft06f001 dd sysout=r
250 /*
260

submit *

```

The output data set and the job management messages produced will be in the CRJE SYSOUT class.

Submitting a Data Set from a User's Library

You can also submit jobs when you are in command mode (as opposed to edit mode), but there will be no active data set. The data sets that make up the input stream must reside in user libraries - either yours or someone else's. If a data set to be submitted is in a library other than your own, you must specify the userid for the library as well as the name of the data set to be used. The userid is enclosed in parentheses, along with the protection key if the data set was assigned a protection key when it was saved.

Suppose the JCL required to compile and execute the FORTRAN program in your CRJE data set FTNPROG is contained in two unprotected data sets in the library of the user with the userid JONES. One data set, named FTNJCL, contains the following three records:

```

//EXAMPLE JOB MSGLEVEL=1,MSGCLASS=R                00000010
// EXEC FORTGCLG                                    00000020
//FORT.SYSIN DD *                                   00000030

```

The other data set, named GOJCL, contains these three records:

```

/*                                                    00000010
//GO.FT06F001 DD SYSOUT=R                            00000020
/*                                                    00000030

```

To submit the JCL with the program, you specify the data sets in the order in which they are to be combined to form an input stream for the operating system. That is,

```
submit ftnjcl(jones),ftnprog,gojcl(jones)
```

Submitting a Data Set List (DLIST)

A convenient way to submit jobs in an input stream that is made up of the contents of several data sets is the use of a data set list, or DSLIST. A data set list is a list of data set names, one per line, in a CRJE-data set with the DSLIST content attribute. When the data set is

specified on a SUBMIT command, the contents of each data set named in the list is included in the input stream.

If you are in edit mode, the data set list may be in the active area, or one of the data sets in the list may be the active data set.

Each line of the data set with the DSLIST attribute contains either an asterisk (*) to indicate the active data set or the name of one CRJE data set that does not have the DSLIST attribute. If a CRJE data set is in a user library other than your own, the name must include the appropriate userid and, if present, the protection key. The format is the same as that for a name specified on the SUBMIT command.

To illustrate the use of a data set list and, at the same time, the specification of job output in the form of an OS data set, a subroutine for the FORTRAN program in the previous FORTRAN example will be created in a new data set. The subroutine will produce output on an OS data set in the CRJE SYSOUT class.

First, a calling sequence to the subroutine must be added to FTNPROG, and the modified program saved under a new name.

```
edit ftnprog,s
i 202,2
202 call exam2 (koeff(1,1))
204
l 200 250
200 2000 FORMAT (' ',I1,19I6)
202 CALL EXAM2 (KOEFF(1,L))
210 STOP
220 END
s ftnrtn1
end
```

Now the subroutine, EXAM2, must be created and saved.

```
edit ftnrtn2,new,s,fortg
10 subroutine exam2 (koeff)
20 dimension koeff(20,20)
30 write (9,3000)
40 3000 format ('1',25x, 'inverted table of binary coefficients' //)
50 do 40 k=1,20
60 i = 21 -k
70 write (9, 2000) (koeff(i,j),j=1,i)
80 40 continue
90 return
100 2000 format (' ',i1, 19i6)
110
120
s
end
```

The JCL in the data set FTNJCL, which is in the library of the user JONES, can still be used, but another DD statement must be added to the GOJCL data set in JONES's library. The new data set will, of course, have to be saved in your own library (not JONES's).

```
edit gojcl (jones)
i 22,2
22 //go.ft09f001 dd unit=2311, disp=(new,catlg),space=(trk,(10,1)),
24 //dsname=brown.example
26
s
end
```

There are now four data sets that compose one job stream. You can create a DSLIST data set containing the names of the four data sets in the order they should be combined.

```
edit ftndslst,new,dslst
  10 ftnjcl (jones)
  20 ftnrtn1
  30 ftnrtn2
  40 gojcl
  50
```

```
s
end
```

Any time you wish to submit the FORTRAN program with the two routines, you may simply enter

```
submit findslst.
```

OBTAINING OUTPUT

When a job finishes execution, you will receive a message indicating normal or abnormal termination. Receiving this message indicates that the output of the job is available for you. The way you obtain job output at your terminal depends upon how the output was specified on the DD statement when the job was submitted.

Take the output created by the FORTRAN program in the preceding example. System messages and one data set were directed to the CRJE SYSOUT class, while the other data set was created as an OS data set - named, cataloged, sequential, and resident on a direct-access storage device.

SYSOUT Output

To obtain the output of the data set from the CRJE SYSOUT class, use the OUTPUT command with the jobname as the operand.

output example

If you want the job management messages returned to you, specify the keyword SMSG also.

Following an OUTPUT command, all the CRJE SYSOUT output for the specified job is returned. Since the job, EXAMPLE, produced only one data set in the CRJE SYSOUT class, that data set is the only one printed. The terminal output is shown in Table 1.

Note that there are no line numbers. The data set is listed from beginning to end. If your terminal does not have a special Interrupt feature, you may have to take positive action to receive each group of "n" output lines, where "n" is installation-defined. The action required is described under Interrupting Output in the Appendix for your terminal type.

You may also discontinue the transmission of output to your terminal. If your terminal is equipped with the Interrupt feature, you can discontinue at any time. If your terminal is not equipped with the Interrupt feature, you can discontinue between groups of "n" lines. The

procedures for this are described under Interrupting Output in the Appendix for your terminal type.

A data set in the CRJE SYSOUT class is deleted after it has been transmitted in its entirety. If you interrupt output while a data set is being transmitted, the data set is not deleted. You may resume receiving the output at your terminal by using the CONTINUE command or reentering the OUTPUT command. You may have the data set and the rest of the CRJE SYSOUT output directed to devices at the central installation by entering another OUTPUT command and asking the central operator to enter a CENOUT command for the job.

The CONTINUE command provides optional starting points for resuming output. You can pick up where you left off by specifying the HERE operand or no operand, since HERE is assumed by default. You may start at the beginning of the data set that was interrupted by specifying the BEGIN operand. The NEXT operand results in the deletion of the data set that was interrupted and resumption of output with the next item for that job in the CRJE SYSOUT class.

Note: If control characters are present in the data set requested by the OUTPUT command, they will be removed before the actual output is transmitted to the terminal.

TABLE OF THE BINOMIAL COEFFICIENTS FOR N FROM 0 TO 19
COMPUTED USING INTEGER ARITHMETIC, WITHOUT COMPUTING ANY FACTORIALS

1																				
1	1																			
1	2	1																		
1	3	3	1																	
1	4	6	4	1																
1	5	10	10	5	1															
1	6	15	20	15	6	1														
1	7	21	35	35	21	7	1													
1	8	28	56	70	56	28	8	1												
1	9	36	84	126	126	84	36	9	1											
1	10	45	120	210	252	210	120	45	10	1										
1	11	55	165	330	462	462	330	165	55	11	1									
1	12	66	220	495	792	924	792	495	220	66	12	1								
1	13	78	286	715	1287	1716	1716	1287	715	286	78	13	1							
1	14	91	364	1001	2002	3003	3432	3003	2002	1001	364	91	14	1						
1	15	105	455	1365	3003	5005	6435	6435	5005	3003	1365	455	105	15	1					
1	16	120	560	1820	4368	8008	11440	12870	11440	8008	4368	1820	560	120	16	1				
1	17	136	680	2380	6188	12376	19448	24310	24310	19448	12376	6188	2380	680	136	17	1			
1	18	153	816	3060	8568	18564	31824	43758	48620	43758	31824	18564	8568	3060	816	153	18	1		
1	19	171	969	3876	11628	27132	50388	75582	92378	92378	75582	50388	27132	11628	3876	969	171	19	1	

END OF DATA

Table 1. CRJE SYSOUT output for job EXAMPLE

You should not use the OUTPUT command with unprintable data such as object modules, for they may contain embedded line control characters. Doing this may cause a TP line error.

The activity of data within CRJE discussed in this major section--creating and editing data sets, submitting jobs and receiving their output--is illustrated in Figure 2. Included in the figure are the commands and subcommands that perform the different activities.

OS Data Sets

To obtain a listing of an OS data set, you must use a combination of EDIT and LIST. In other words, the OS data set must be copied into the active area, where it becomes your active data set. As mentioned before, the data set must be named, cataloged, and resident on a single DASD volume that was mounted at system start-up. Either a sequential data set or a member of a partitioned data set may be copied into the active area.

As records of an OS data set are copied into the active area, line numbers are assigned. Each logical record is transformed into an integral number of 80-byte records, with blanks padded to the right in the last line of each set, if necessary. Line numbers are assigned in increments of 10, starting with line number 10.

Attributes are also assigned to the active data set. The line increment attribute is 10, and the content and sequence attributes default to DATA and SEQ unless specified otherwise on the EDIT command. The NOSEQ sequence attribute should be assigned by specifying the NONUM operand, since a line number in positions 73-80 might overlay part of a record.

To list the data set named BROWN.EXAMPLE, which was produced by the FORTRAN program, specify the data set name, enclosed by single quotes, on the EDIT command. The LIST subcommand with no operand causes the entire data set to be listed. You may also request that specific lines be listed.

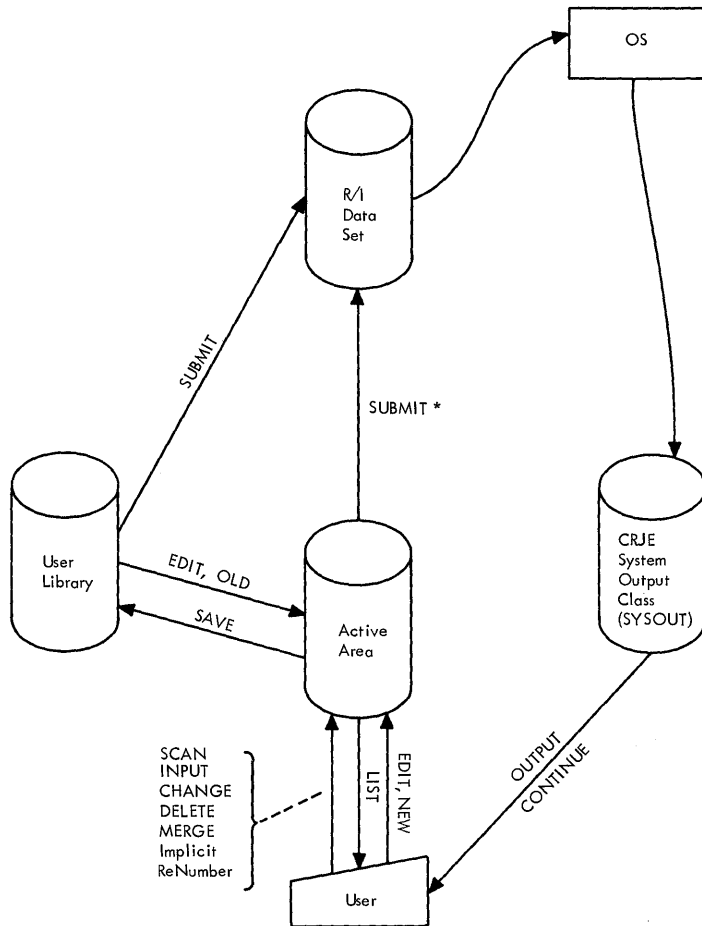
```
edit 'brown.example',nonum
```

```
list
```

If you interrupt output while you are listing the active data set, you must reenter the LIST subcommand to resume the listing. You can specify the starting point by giving line numbers.

OS data sets are not automatically deleted at any time by CRJE. If, however, the first qualifier of the data set name is your userid, you can delete the data set by using the DELETE command.

```
delete 'brown.example'
```



(Note: The difference between the two SUBMIT commands is that SUBMIT alone allows the submission of any data set in a user's library, whereas SUBMIT * allows the submission of the active data set for job processing. SUBMIT functions as an EDIT subcommand when used with the active data set.)

Figure 2. CRJE Data Activity

CANCELING JOBS

You can cancel a job you have submitted any time the job is still in the system - that is, from the time you submitted it until the last CRJE SYSOUT output is returned to you. This is done using the CANCEL command with the jobname as the operand. CANCEL can be entered in either command mode or edit mode.

cancel example

There may be a delay if you cancel the job before it has begun execution. Be sure you receive the message

JOB CANCELLED EXAMPLE

before submitting another job with the same name.

OBTAINING STATUS INFORMATION

There are commands you can enter to request information about jobs you have submitted, the data sets in your library, or a particular data set in your library.

JOB STATUS

The STATUS command is used to request information about jobs you have submitted. If you want the status of one particular job, specify the jobname in the operand field. If you would like a list of all the jobs you have submitted that are still in the system, leave the operand field blank.

Assume you have three jobs - JOB1, JOB2, and JOB3 - in the system. JOB1 has completed, JOB2 is currently executing, and JOB3 is scheduled first in the queue of jobs waiting to be executed. The STATUS command and the response will look like this:

```
status
STATUS      JOB1      NORMAL END
STATUS      JOB2      EXECUTING
STATUS      JOB3      SCHED(1)
```

DATA SET STATUS

The LISTLIB and LISTDS commands are used to obtain data set status. If you want a list of all the data sets in your library and the status of each, use the LISTLIB command. If you want information about one particular data set in your library, use the LISTDS command and specify the data set name first in the operand field.

The information returned in response to either command always includes the data set name, the content attribute of the data set, and an indication of the sequence attribute of the data set. There are two optional operands on each command to request additional information. The STATUS operand, abbreviated S, requests the number of blocks of 10 lines each that are in the data set and an indication of whether or not the data set is protected. The HISTORY operand, abbreviated H, requests the creation date of the data set, the last time it was changed, and the number of times it has been used in CRJE operations.

The information requested by a LISTLIB or LISTDS command is returned in columns, each with a heading describing the information in that column. The following three headings are always given: DSNAME, the simple name of the data set; TYPE, the content attribute of the data set; and LNO, YES for the SEQ sequence attribute or NO for the NOSEQ attribute. If the STATUS operand is specified, the following two headings are provided: BLOCKS, for the number of 10-line blocks in the data set; and KEY, YES if protected or NO if unprotected. The other three possible headings are included if the HISTORY operand is specified. They are: CREATED, the date first saved; LAST CHG, the date last saved; and XREAD, the number of times used in CRJE processing since created. All dates are given in the form yy.ddd, where yy is the year and ddd is the day of the year.

Note: Attributes assigned to non-CRJE data sets that are listed via LISTLIB or LISTDS are unreliable until the data sets are edited. The

type or content attribute is established with the EDIT command. Other data set information is supplied when the data sets are resaved.

For example, the LISTLIB command and the responses would look like this, assuming your library contains six data sets:

```
listlib
DSNAME      TYPE      LNO
MYDATA      DATA     YES
FTNPROG     FORT-G    YES
FTNRTN1     FORT-G    YES
FTNRTN2     FORT-G    YES
GOJCL       DATA     YES
FTNDSLST    DSLIST   YES
```

If you request more information about a particular data set, your session might go as follows:

```
listds mydata,s,h
DSNAME      TYPE      LNO  BLOCKS  KEY  CREATED  LAST  CHG  XREAD
MYDATA      DATA     YES   2        NO  69.120  69.140  25
```

```
listds ftndslst,s
DSNAME      TYPE      LNO  BLOCKS  KEY
FTNDSLST    DSLIST   YES   1        NO
```

SENDING MESSAGES

You may send a message to the computer operator at the central installation or to another terminal user. The SEND command is used for sending messages. It may be entered in command mode or edit mode.

CENTRAL OPERATOR

To send a message to the central operator, simply enter the SEND command and the text of the message. The text is limited to 40 characters and must be enclosed in quotes unless it does not contain blanks, parentheses, or commas. If the text contains a single quote (e.g., an apostrophe), you must enter it as two single quotes (which will be counted as one character).

```
send 'isn't volume 666666 mounted?'
```

The operator will receive the following message:

```
CRJE MSG FR BROWN ISN'T VOLUME 666666 MOUNTED?
```

If your central installation uses multiple consoles, check the SEND command description in the reference part of this publication for how to specify a particular console type.

TERMINAL USER

To send a message to another terminal user, specify the message text first in the operand field, followed by the USERID, or U, keyword with the userid of the recipient in parentheses.

```
send 'did your job run successfully?' u(jones)
```

JONES, if he is active, will receive the following message:

```
CRJE MSG FR BROWN DID YOUR JOB RUN SUCCESSFULLY?
```

If JONES is not active, the message will not be delivered, and you will be notified as follows:

```
USER INACTIVE, MSG NOT SAVED.
```

If you want a message to be saved for later delivery if the recipient is inactive, you may specify the LOGON operand, abbreviated L, on the command.

USING A COMMAND LIST (CLIST)

CRJE provides a special facility that allows you to keep an often-used series of commands in a data set and to execute the whole series by entering only one command, the EXEC command.

The series of commands is called a command list, or CLIST, and the content attribute of the data set containing the command list must be CLIST.

CREATING A CLIST

Terminal commands and subcommands may be included in the command list, but they cannot be continued from one line to the next. The only commands that are not allowed are LOGON, LOGOFF, and EXEC.

The CLIST attribute must be assigned when the data set is created. Specifying NONUM will allow 80 characters per line rather than 72.

The example that follows illustrates a series of commands and subcommands that combine the two FORTRAN routines introduced earlier into one program, which is then submitted for execution.

```
edit fortran,new,clist,nonum
  10 edit ftnprog
  20 merge ftnrtn2 30,90 200
  30 submit ftnjcl (jones),*,gojcl
  40 end
  50
```

```
save
```

```
end
```

EXECUTING A CLIST

Whenever you wish to submit the FORTRAN program composed of the two routines, you simply enter

```
exec fortran
```

The series of commands and subcommands will be processed one by one in the order they appear in the command list.

If you would like the command or subcommand to be printed at your terminal when it is processed, you may specify the LIST operand, abbreviated L.

CORRECTING A CLIST

If a command or subcommand in a CLIST contains an error that would require it to be reentered, execution of the command list is terminated. Any commands before the one in error will have been processed and jobs already submitted are not affected. If a command or subcommand requires input from you, you will be prompted in the normal way and can enter the necessary information.

INTERRUPTING OUTPUT DURING CLIST EXECUTION

If, at any time during the execution of a CLIST, you interrupt the transmission of output to your terminal, you have a choice of resuming execution with the next command in the CLIST or of entering the next command from your terminal.

To interrupt output and resume execution with the next command in the CLIST, just enter

end

This use of END does not terminate edit mode.

If no more of the CLIST is to be executed, enter the next command to be processed. Remember that you are in whatever mode you were in as a result of the last command processed in the CLIST.

Note: If 1) the automatic break feature is used to terminate CLIST execution and 2) a CLIST command is the last line listed at the terminal when execution ends, then the command itself is not executed.

COMMAND FORMAT

The general format of command language statements is:

Operation	Operand
A verb or the abbreviation of a subcommand verb.	One or more operands separated by delimiters.

The operation field contains a verb that identifies the command or subcommand and the associated action. Subcommand verbs may be abbreviated. There is no verb in the case of the Implicit subcommand, since the action to be taken is implied by the operands specified.

The operand field may contain one or more operands or, for some commands and subcommands, may be blank. The operands either provide required information or request optional system actions. They may be positional or keyword. A positional operand consists of a symbolic name (e.g., dsname or jobname), a character string (a message text, for example), or one or two numbers (to specify line numbers and increment values). Positional operands must be first in the operand field and must appear in the prescribed order. In the format illustrations, these operands are represented in lowercase letters to indicate that the user must substitute specific information or specific values.

Keyword operands may appear in any order following the positional operands. They are distinguished by a keyword, which in the illustrations is shown in uppercase characters to indicate it must be entered exactly as shown. A keyword operand may be followed by a parameter that must be enclosed in parentheses. For example, in the operand ACCT (accounting information), ACCT is the keyword followed by the parameter, "accounting information", for which the user substitutes his actual accounting information.

Although the illustrations show upper and lowercase characters, the commands may be entered using either upper or lowercase alphabets. Lowercase alphabets in commands are automatically translated to upper case by the system.

DELIMITERS

Delimiters are used to separate the operation field from the operand field and to separate operands in the operand field. One or more blanks must be used between the operation and operand fields. Generally, delimiters between operands may be one or more blanks, a single comma, or a combination of a single comma and one or more blanks. In the format illustration, a blank between operands indicates that any of these delimiters may be used. In special cases, a delimiter must be a single character, which may vary according to the command in which it is used and the operands it separates. A single character delimiter is represented by the symbol Δ.

CONTINUATION

If necessary, a command or subcommand, with the exception of the Implicit subcommand, may be continued from one line to the next. The division can be made between the operation field and the operand field or between operands in the operand field. To indicate that the next line is a continuation line, the user must enter a hyphen (-) immediately preceding the end-of-line sequence for the current line. The first operand in the continuation line may be preceded by delimiters.

Terminal commands added to the system by the installation cannot be continued.

Terminal commands in a CLIST data set cannot be continued.

COMMAND NOTATION

The terminal command descriptions are presented according to the following subjects: session management, data set management and data manipulation, job submission and output retrieval, status information, messages, and command list execution. Each description includes the following:

1. The command name.
2. A brief functional description.
3. The command format illustration.
4. A description of the operands.
5. A description of the command, discussing what the command does, explaining any restrictions and limitations, and providing programming notes.
6. Examples.

FORMAT ILLUSTRATION CONVENTIONS

Several conventions are followed in illustrating the format of the terminal command language:

- Uppercase letters, numbers, and special characters must be entered exactly as shown. Exceptions to this convention are brackets, []; braces, { }; ellipses, ...; subscripts, and Δ . These are never coded.
- Letters and words in lower case represent variables for which you must substitute specific information or specific values.
- Items within brackets are optional; they may be omitted at your discretion. Any item not within brackets must be coded. For example, take the format illustration:

```
LISTDS dsname [S[TATUS]] [H[HISTORY]].
```

The verb LISTDS must be entered as shown. The data set name is the only required operand. Both STATUS and HISTORY are optional operands and may be abbreviated as S and H, respectively.

- Stacked items represent alternative items. No more than one of the stacked items may be entered. If the items are enclosed in braces, i.e., {dsname}*,

one of the items must be specified.

- If stacked items are enclosed in brackets, you have the option to specify one of the items or to specify nothing. For example, the

```
[ON ]
[OFF]
```

operand of the SCAN subcommand may be omitted entirely so that the automatic scan will not be affected, or either ON or OFF can be specified to turn the automatic scan on or off.

- If one of a group of stacked items enclosed in brackets is underlined, that item is implied, even though you may not specify anything. For instance, if you do not specify either item of the

```
[BC ]
[NOBC]
```

operand of the LOGON command, you will receive the broadcast messages by default.

- An ellipsis, ..., indicates that the preceding item or group of items can be entered more than once in succession. For example, SUBMIT dsname ... means that a series of data set names can be specified.

TERMINOLOGY

Following is a list of terms and their definitions used in the command descriptions.

dsname

The name of a data set. The name has several formats, depending upon whether it identifies a CRJE data set or an OS data set.

dsname [(userid [key])]

This is the format for a CRJE data set name. The dsname, userid, and key symbols are separated using the same delimiters that are valid between operands. The userid and key specifications must be enclosed in parentheses.

dsname: A 1-8 character name that is assigned to a CRJE data set. The name may contain the alphameric characters A-Z, 0-9, @, #, and \$. The first character must be alphabetic. This corresponds to a member name in the user's library.

userid: The 1-7 character identification of the user in whose library the data set resides.

key: A three-character protection key that can be assigned to a CRJE data set.

'dsname[(member name)]'

This is the format for an OS data set name. The dsname and member name symbols cannot be separated by blanks or commas. The entire operand must be enclosed in single quotes. The member name must be enclosed in parentheses.

dsname: The full name, including qualifiers, of the OS data set. If the data set is to be deleted with the DELETE command, the first qualifier must be the userid of the user entering the command.

member name: The simple name, enclosed in parentheses, that specifies a member of a partitioned data set.

increment

The numeric value used to develop successive line numbers in a data set. A value of 10 is assumed when no increment is specified.

jobname

The symbol specified in the name field of a JCL JOB statement. This consists of 1-8 alphameric characters, the first of which must be alphabetic.

key

A three-character key assigned to a CRJE data set for protection. The characters can be alphabetic, numeric, or special.

linenum

Line number: a 1-8 character number that specifies the number associated with an existing line of input or that specifies the number to be assigned to a line currently being entered.

password

An identifier used for the password, consisting of 1-8 alphameric characters, the first of which must be alphabetic. A password is assigned to each user by the central installation.

userid

The identification assigned to a user. This consists of 1-7 alphameric characters, the first of which must be alphabetic.

SESSION MANAGEMENT COMMANDS

LOGON

The LOGON command identifies the user and initiates his session.

Operation	Operand
LOGON	userid/password [A[CCT](accounting information)]
	[BC] [M[SGID] [NOBC] [NOM[SGID]]

userid/password

Specifies the user's identification code and password, which are assigned to the user by the installation. The userid consists of 1-7 alphameric characters, and the password consists of 1-8 alphameric characters. If either is invalid or the userid is currently being used, the user will receive a message requesting the correct information. If acceptable information is not

supplied, the LOGON is rejected and, for switched connections (i.e., established by dialing), the line is disconnected.

A[CCT](accounting information)

Specifies accounting information, which consists of parameters passed to an accounting routine provided by the installation. The parameters are delimited by blanks and/or a single comma, and the entire field is enclosed in parentheses. Up to 33 characters, not counting delimiters, may be specified. If blanks or commas are included in a parameter, the parameter must be enclosed in single quotes, e.g., ACCT(AA,'B,B' 'C C'). If an accounting routine is not provided by the installation, this operand is ignored.

BC

Requests a copy of the broadcast messages. Broadcast messages may also be requested with the LISTBC command.

NOBC

Specifies that the broadcast messages not be listed. If this operand is omitted, the broadcast messages will be listed.

M[SGID]

Specifies that message codes, or identifiers, are to be printed with terminal messages that are self-explanatory in the context of the session dialogue. The codes are not printed if this operand is omitted.

NOM[SGID]

Specifies that message codes are not to be printed with terminal messages that are self-explanatory in the context of the session dialogue. This is the default option.

The LOGON command must be the first command the user enters. If the terminal is inactive, the user must first enter a null line in order to receive an underscore prompt, which indicates the system is ready to accept the command.

If another user is logged on and in command mode, a LOGON command automatically logs him off and logs the new user on. This procedure is advantageous when operating with a switched line connection, since the line is not disconnected between sessions.

Examples

1. logon jones/ralph

In this example, JONES is the userid and RALPH is the password. These are the only two operands required by CRJE. After entering a command of this form, the user will receive the broadcast messages in addition to any messages generated for him while he was inactive. System messages sent to him during this session will not have message codes appended to them.

2. logon jones/ralph, a(138303), nobc, msgid

In this example, the user JONES logs on with the account number, 138303, which will be checked by an installation-provided routine, if one is provided. Broadcast messages will not be sent, and message codes will be appended to system messages sent to the user.

Specifies the name of the CRJE data set to be created, or the name of a CRJE or OS data set to be edited. This operand is required and must be the first operand entered. If a data set is to be created, the operand NEW must also be specified. If the operand OLD is specified or if neither NEW nor OLD is specified, the data set is to be edited. If it does not exist, the user will be notified. The format of dsname depends upon whether the data set is a CRJE or OS data set.

dsname [(userid [key])]

Specifies the CRJE data set to be created or updated.

dsname: specifies the simple name of a data set to be created or an existing data set in the library of the user entering the command.

dsname (userid): specifies the name of an unprotected data set that exists in the library of the user identified by the userid.

dsname (userid key): specifies the name of a protected data set that exists in the library of the user identified by the userid. Key specifies the protection key assigned to the data set.

'dsname[(member name)]'

Specifies the fully qualified name of an OS data set residing on a single volume of a direct-access storage device. This volume must have been mounted at the last CRJE system start-up. Only cataloged, sequential data sets and members of cataloged, partitioned data sets are valid. In order to protect the CRJE system library and CRJE user libraries, any dsname beginning with CRJE. is rejected.

NEW

Specifies that a data set is to be created. The dsname operand must be the simple name of a CRJE data set.

OLD

Specifies that a data set is to be edited. The dsname operand must specify an existing CRJE or OS data set.

NUM

Specifies that line numbers are to be maintained as the last eight characters of lines entered in input mode or with the Implicit subcommand. The user will therefore be restricted to 72 characters of data input per line.

This operand assigns the SEQ sequence attribute to a new data set. The data set retains this attribute unless specifically indicated otherwise when the data set is subsequently edited. If specified for an existing CRJE data set with the NOSEQ attribute, the attribute is changed although only subsequently entered lines will actually contain line numbers.

This operand should not be specified for an OS data set. If specified, the last 8 characters of each record in the active data set are overlaid with a line number, and the records will be treated as 80-character records.

NONUM

Specifies that line numbers are not to be maintained as the last eight characters of lines entered in input mode or with the Implicit subcommand. If this operand is specified for a CRJE data

set, the data set is assigned the NOSEQ sequence attribute. This attribute is not changed unless NUM is specified in a subsequent EDIT command issued for the data set.

This operand should always be specified when editing an OS data set.

S[CAN]

Specifies that the automatic syntax scan by statement is to be invoked for all lines subsequently entered in input mode. If this operand is omitted, input lines will not be scanned.

NOS[CAN]

Specifies that lines subsequently entered in input mode are not to be scanned for syntax errors. This is the default option.

PL1

FORT { E
 G
 H }

DSL
LIST
CLIST
DATA
TEXT

Specifies the content attribute to be assigned to the data set. If this operand is omitted and either a CRJE data set is to be created (i.e., NEW is specified) or an OS data set is to be edited, the DATA attribute is assigned by default. An attribute specified for an existing CRJE data set replaces the currently assigned attribute of the data set when it is saved under the same name. The content attribute specified for formerly non-CRJE data sets is unreliable unless established by the user. It may be set as specified with the EDIT command. The attributes that can be assigned are as follows:

PL1 ([integer1 integer2] [C[HAR]48]
 [C[HAR]60])

Indicates PL/I source language statements. These statements can be scanned for syntax errors if the installation included the PL/I Syntax Checker in the system. Lowercase alphabetic characters entered in source statements entered for this data set will be translated to upper case.

C[HAR]60 and C[HAR]48 specify whether the 60-character set or the 48-character set is used in source statements. C[HAR]60 is the default option.

Integer1 specifies the left source margin. It must be less than the right source margin. integer2 specifies the right source margin. It must not exceed 72 if line numbers are maintained in positions 73-80. The default option is 2 (left source margin) and 72 (right source margin). Zero (0) may not be specified for either integer1 or integer2. If either source margin is incorrectly specified, message IHK356 INVALID SOURCE MARGINS is sent to the user.

FORT { E }
 { G }
 { H }

Indicates FORTRAN IV source language statements. These statements can be scanned for syntax errors if the installation included the FORTRAN Syntax Checker in the system. E, G, and H indicate the language level to be scanned. If the level specified is not supported, no syntax checking will be performed. Lowercase alphabetic characters entered in source statements entered will be translated to upper case.

DSLISIT

Indicates a data set list, which consists of the names of CRJE data sets. Each line of the DSLISIT data set contains only the name of one CRJE data set, which cannot have the DSLISIT attribute. The name has the format dsname [(userid [key])]. When a data set with this attribute is specified in a SUBMIT command, the contents of all data sets named in the list are entered into the OS input stream.

CLIST

Indicates a command list, which is a list of terminal commands and subcommands, with the exception of LOGON, LOGOFF, and EXEC. Each line of the CLIST data set contains a command or subcommand, just as it would be entered from the terminal for processing. When a CLIST data set is specified in an EXEC command, each command or subcommand in the data set is executed as if it had been entered from the terminal.

DATA

Indicates undefined data in which alphabetic characters are to be maintained in upper case. Lowercase alphabetic characters entered in input mode or with the Implicit subcommand will be translated to upper case.

TEXT

Indicates undefined data in which alphabetic characters are to be maintained in upper and lower case, exactly as entered.

From the time the user enters the EDIT command to create an active data set until he enters the END subcommand to delete the active data set, he operates in one of two modes: input mode or edit mode. The modes are distinguished by the type of input the user can enter. In input mode, he enters input lines that become records of the active data set. The content attribute of the data set describes the type of records it contains. Thus, when creating a data set with the CLIST attribute, the lines entered in input mode are actually commands. While in input mode, the user is prompted with a line number whenever he can enter a line of input. These line number prompts may be suppressed by using the NOPROMPT operand of the INPUT subcommand.

In edit mode, the user can enter EDIT subcommands, which request editing functions for the active data set. There are EDIT subcommands for inserting, replacing, deleting, and changing lines of the active data set. Lines can be listed, renumbered and syntax checked. The user can merge data sets and can set tabs for formatting his input and output. The user must enter the SAVE subcommand prior to entering the END subcommand if he wants the active data set to become a part of his permanent user library. While in edit mode, the user is prompted with an underscore whenever he can enter a subcommand.

To switch from one mode to the other, the user simply enters a null line in response to the prompt. If a null line is entered in response to a line number prompt, the user receives an underscore prompt and can

enter a subcommand. If a null line is entered following an underscore prompt, the user receives a line number prompt for the line following the last existing line in the data set.

Creating a CRJE Data Set

When creating a data set, the dsname and NEW operands must be specified on the EDIT command. The user must also specify those operands that indicate attributes for the data set unless the proper attributes are assigned by default. The line increment attribute, which determines the difference between line numbers, is always 10 for a new data set. To change the line increment attribute, the user must issue the RENUMBER subcommand. The content attribute and sequence attribute are assigned according to operands specified. If these attributes are not specified, the DATA and SEQ attributes are assumed.

If the data set has either the FORT or PL1 content attribute, the user can request an automatic scan of source statements entered in input mode (but not with the Implicit subcommand). To request the automatic scan the user must specify the S[CAN] operand.

After an EDIT command with the NEW operand is entered, the user is in input mode and receives a prompt for line number 10. He continues to enter input lines in response to the line number prompts until he enters a null line, which puts him in edit mode so he can enter EDIT subcommands. He can resume input mode from the point at which he stopped by entering a null line in response to an underscore prompt that is received in edit mode.

When the data set is complete, the user can save it in his permanent user library or can include it in a job entry. To indicate he is through with the active data set, he must enter the END subcommand. The active data set is then deleted and the user returns to command mode.

Examples

1. `edit mydata new`

This command initiates the creation of a CRJE data set, which is named MYDATA. The content attribute is DATA, so lowercase characters in all lines entered will be translated to upper case. The sequence attribute is SEQ, so line numbers will be maintained in positions 703-80 of each line entered. Line numbering will start at 10 and will proceed in increments of 10, the line increment attribute.

2. `edit myprog new pl1 s`

When the command is entered, a new data set named MYPROG will be created. The attributes are PL1 (content), SEQ (sequence), and 10 (line increment). PL1 source statements will be automatically scanned for syntax errors as they are entered.

Editing a CRJE Data Set

Since the OLD operand is the default option on EDIT, the only required operand is the name of the CRJE data set to be edited. The data set must be in a user library. Only the simple name (i.e., the one specified when the data set was created) is required if the data set is in the library of the user entering the command. If it is in another user's library, that user's userid must also be specified, and if it is protected, the protection key must be included.

If the user specifies an operand that assigns a content or sequence attribute, that attribute is assigned to the active data set and replaces the one currently assigned if the data set is saved under the same name. If the sequence attribute is changed, positions 73-80 of existing lines are not modified, even though these lines will subsequently be handled according to the new attribute.

When an EDIT command with the OLD operand is entered, the data set specified is copied into the active area to serve as the active data set. The user enters edit mode and receives an underscore prompt for an EDIT subcommand. If he wants to add lines at the end of the data set, he can enter input mode by entering a null line following the underscore prompt, after which he receives a prompt for the line following the last existing line in the data set.

After the user has finished making the desired modifications, he can save the updated version in his user library by using the SAVE subcommand. The updated version will replace the original copy in the user library unless a different name is specified on the SAVE subcommand. When he is through with the active data set he must enter the END subcommand to return to command mode.

Note: For non-CRJE data sets -- If the user enters all x's in the first eight positions of a record in a non-CRJE data set, EDIT will sequence the data set, replacing the x's with zeros. EDIT checks to see if these eight characters are numerics or zeros; if they are not, message IHK402 -- LINE NUMBER NOT xxxxxxxx OR NUMERICS line number -- is sent to the user. This message will appear only once, even if each record in the data set contains invalid characters in the first eight positions. Furthermore, if a non-CRJE data set has the SEQ attribute, the line numbers that are generated by the x's in the first eight positions will also be stored in positions 73-80 of the record.

Examples

1. `edit myprog,s`

This command initiates editing functions for the data set named MYPROG, which is in the library of the user entering the command. A copy of MYPROG becomes the active data set. The attributes, which are PL1, SEQ, and 10, are not changed. Any source in input mode while MYPROG is the active data set will be automatically scanned for syntax errors.

2. `edit sample (jackson)`

In this example, SAMPLE is the name of a data set in the library of the user whose userid is JACKSON. The user, JACKSON, had not assigned a protection key to SAMPLE. A copy of SAMPLE now serves as the active data set. If the user enters the SAVE subcommand, the active data set will be put in his own user library (not JACKSON's library) with the name SAMPLE. The user will be prompted for a new dsname if his library already contains a data set named SAMPLE.

Editing an OS Data Set

An OS sequential data set or a member of an OS partitioned data set can be copied into the active area to serve as the active data set. The user specifies the fully qualified name of the data set, which must be cataloged and resident on a single volume of a direct-access storage device. The member name must be specified in parentheses for a partitioned data set, and in all cases, the entire dsname operand must be enclosed in single quotes.

Note: In order to protect the CRJE system library and CRJE user libraries. any dsname of an OS data set beginning with CRJE. is rejected.

Attributes are assigned when the data set or member is copied into the active area. DATA is assumed if no content attribute is specified. The line increment attribute is 10, and line numbers are automatically assigned, starting with line number 10. If the logical record size of the OS data set is greater than 72 and if the SEQ attribute is specified or assumed; i.e., neither NUM nor NONUM is specified, each record is transformed into an integral number of 72-byte records, with blanks padded to the right in the last record of each set, if necessary. The NOSEQ sequence attribute should be assigned by specifying the NONUM operand; each record will be transformed into an integral number of 80-byte records, with blanks padded to the right in the last record of each set, if necessary. For fixed-length, blocked or unblocked records, if the length of the logical records does not exceed 120 bytes, an entire record is printed on one line when the data set is listed.

Edit mode is entered when an OS data set is specified, and the user can list or modify the data set using EDIT subcommands. If the user saves the active data set in his library, he must specify a simple dsname for it on the SAVE subcommand.

Example

1. `edit 'jones.result*(phase1)' nonum`

As a result of this command, the member PHASE1 of the partitioned data set JONES.RESULTS is copied into the active area. The attributes of the active data set are DATA (content), NOSEQ (sequence), and 10 (line increment).

Editing a Data Set After Abnormal Session Termination

If a user's session is terminated abnormally while he has an active data set, CRJE attempts to save the active data set in the user's library. If it is successfully saved, then the next time the user logs on, he receives the message, SESSION TERMINATED - ACTIVE AREA SAVED -DSNAME=ACTIVE. the data set active is then treated as an existing CRJE data set, except that attributes must be reassigned.

Example

1. `edit active,old, forte`

This command initiates editing of the data set that was saved when the user's session terminated abnormally. The FORTE attribute must be reassigned, even though it is an existing (OLD) data set. The line increment attribute is 10, and the sequence attribute is SEQ.

EDIT Subcommands

INPUT: The INPUT subcommand allows the user to enter lines of input into the active data set. The user can specify replacement only, insertion only, or a combination of the two. Input mode is entered as a result of the INPUT subcommand.

Operation	Operand
I[INPUT]	[linenum [[increment] [I]]] [P[ROMPT] NOP[ROMPT]]

linenum

Specifies the line number of the first line to be entered in input mode. Line numbering begins with the specified number and, unless otherwise specified, will continue in increments of whatever line increment attribute is assigned to the data set. If neither the I nor the R operand is specified, there is no distinction between insertion and replacement. If a line number exists, the existing line is replaced by the line that is entered. If a line number does not exist, the input line is inserted following the the line with the line number that is less than that of the line to be inserted.

increment

Specifies the increment between those lines entered in input mode immediately following the INPUT subcommand. This increment is used only temporarily and does not become an attribute of the data set. This operand is not applicable when the user requests replacement only by specifying the R operand.

I

Specifies that the lines about to be entered are to be inserted only, beginning with the line number specified in the subcommand. An increment of 1 is used unless another increment is specified. If a line number is generated that is greater than or equal to an existing line number, the user will receive an error message.

R

Specifies that the lines about to be entered are to replace existing lines, starting with the line number specified in the subcommand. If the specified line does not exist, the user will receive an error message. The line number prompts the user will receive correspond to the existing lines in the data set.

P[ROMPT]

Specifies that a line number prompt is to be printed prior to each line entered in input mode following this subcommand.

NOP[ROMPT]

Specifies that line number prompts are not to be printed for lines entered in input mode. If this operand is omitted, the line number prompts will be printed. The lines entered subsequently are numbered, even though the line numbers are not printed.

Note for 2740, 2741, and 1050 users: If you desire to have your lines of input appear on separate lines, you will need to enter a carriage return after each line entered when NOPROMPT is specified, since the system will not generate a carriage return.

When an INPUT subcommand is specified, input mode is entered, during which the user types in lines that become records of the active data set. If no line number is specified in the subcommand, line numbering begins following the last line of the active data set, using the line increment attribute of the data set.

When the user has finished entering lines, he can terminate input mode by entering a null line in response to the next line number prompt. He then resumes operation in edit mode. The operands of the INPUT subcommand will not be carried over to the next time input mode is entered.

Examples

1. i

Following this subcommand, the user will receive a line number prompt for the line following the last line in the active data set. For example, if the last line in the active data set is number 150 and the line increment attribute is 10, then the first prompt will be 160, followed by 170, and so on.

2. input 100 5 nop

This form of the subcommand causes numbering of input lines to start with 100 and proceed in increments of 5. Line numbers are assigned to the input lines, although no line number prompt is printed prior to the entry of each line. An input line replaces any existing line that has the same line number, e.g., line 100 replaces an existing line 100. The input line is inserted if no line with the same number exists, e.g., line 105 is inserted between lines 100 and 110 if no line 105 exists.

3. i 82,2, i

This subcommand specifies that lines are to be inserted in the active data set, starting with line number 82 and continuing in increments of two. No existing lines will be replaced. If the user does not terminate input mode before a prompt is generated for an existing line number, then input mode is automatically terminated and the user is notified of the error. For instance, assuming that 90 is the next line in the data set, after the user enters line 88, the system sends a message instead of a prompt for line 90.

4. i 10,r

Assuming line 10 exists, the first prompt the user receives after this subcommand is 10, and the line the user enters replaces the existing line. Prompting continues with the number of the next existing line, which may be 11, 15, 20, or any other number. The user indicates he has finished replacing lines by entering a null line, which returns him to edit mode.

DELETE: This subcommand is used to remove a line or a group of lines in the active data set.

Operation	Operand
D[DELETE]	[[linenum [linenum]]]

no operand

Specifies that the last line of the active data set is to be deleted.

linenum [linenum]

Specifies the line or the group of lines to be deleted. If a range of lines is specified, those lines with line numbers \geq the first line number specified and \leq the last line number specified are deleted.

Examples

1. d

The last line of the active data set is deleted. The last line is the line with the highest line number.

2. d 100, 150

When this subcommand is entered, all lines between, and including, 100 and 150 are deleted. Lines 100 and 150 do not have to exist.

IMPLICIT: The Implicit subcommand can be used to enter or delete a single line in the active data set while in edit mode.

Operation	Operand
(no operation verb)	linenum[Δtext]

linenum

Specifies the line number of the line to be deleted or entered. If this is the only operand, the line with the specified line number is deleted. If accompanied by the text operand, the line number specified is assigned to that line of text.

Δtext

Specifies the text to be entered as the line with the line number specified in the preceding operand. If a line with the specified line number exists, it is replaced. If one does not exist, the line is inserted. The delimiter Δ must be a single blank or a single comma. The text can consist of up to 80 characters (or 72 if the data set has the SEQ attribute).

The Implicit subcommand allows the user to enter lines in the active data set without being in input mode. Only one line can be entered at a time. No line number prompts are received. The user is not notified whether a line has been inserted or whether it has replaced an existing line.

Examples

1. 50

Line 50 in the active data set is deleted.

2. 80, DO 20 I=2,19

This example illustrates the insertion or the replacement of line 80. If line 80 does not exist, then the statement, " DO 20 I = 2, 19", is inserted at line 80. If line 80 exists, it is replaced by the new statement. Note that the first six characters in the statement are blanks, since the comma is the single character delimiter.

CHANGE: The CHANGE subcommand is used to replace a character string in a line or group of lines of the active data set.

Operation	Operand
C[CHANGE]	linenum [linenum] Δ text1 Δ text2 Δ [A[LL]]

linenum [linenum]

Specifies the line or the range of lines to be affected by the change. If a range is specified, those lines with line numbers ≥ the first line specified and ≤ the last line number specified are affected.

Δtext1Δ text2Δ

Specifies the character string to be replaced - text1 - and the replacement characters - text2. Δ is a single character that delimits the character strings. Δ can be any printable, nonnumeric character other than a comma. Obviously, Δ cannot appear in either character string. Each character string can contain up to 40 characters. If text2 contains more characters than text1, the line will be expanded to accommodate the additional characters. If expansion results in a line of more than 80 characters, the line will be truncated after position 72 for a data set with the SEQ attribute, or after position 80 for a data set with the NOSEQ attribute. The user will be notified if truncation occurs. If text1 is longer than text2, the line is compressed. Positions 73-80 are not affected if the data set has the SEQ attribute.

A[LL]

Specifies that every occurrence of text1 is to be replaced by text2. If ALL is not specified, only the first instance of text1 in each line is replaced.

Note: If a dash is used as a delimiter in this subcommand, it must be followed by a blank or by the ALL operand. Otherwise, the system interprets the dash as indicating a continuation line and will send the message, IHK314 DELIMITER MISSING.

All of the operands of the CHANGE subcommand are positional and must be entered as they appear in the illustration.

Examples

- 1. c 50, 70 /46/3/

This subcommand causes the character 3 to replace the characters 46 the first time they appear in each line from line number 50 through line number 70 of the active data set.

line 50 (before): A46AAA46AAAAA
line 50 (after): A3AAA46AAAAA
line 60 (before): BBB4646BBBBB
line 60 (after): BBB346BBBBB
line 70 (before): 46CCCCCCC46C
line 70 (after): 3CCCCCCC46C

- 2. change 50 *46*1234*all

This command causes every occurrence of the characters 46 in line number 50 to be replaced by the characters 1234.

line 50 (before): A46AAA46AAAAA
line 50 (after): A1234AAA1234AAAAA

MERGE: The MERGE subcommand is used to combine a CRJE data set with the active data set or to copy lines from one place to another within the active data set.

Operation	Operand
M[ERGE]	{ dsname } [linenum linenum] [linenum]
	{ * }

dsname

Specifies the CRJE data set to be combined with the active data set. This operand must have one of the following formats:

dsname: specifies a data set in the library of the user entering the command.

dsname (userid): specifies the name of an unprotected data set in the library of the user identified by the userid.

dsname (userid key): specifies the name protected data set in the library of the user identified by the userid. Key specifies the protection key assigned to the data set.

*

Specifies the active data set. This operand indicates that lines are to be copied from one place to another in the active data set.

linenum linenum

Specifies the range of lines to be inserted in the active data set. Those lines with line numbers \geq the first line number specified and \leq the last line number specified are inserted. If no line numbers are specified, the entire data set is inserted. This operand is positional in that, if entered, it must be entered following the dsname of * operand.

linenum

Specifies the line number of the active data set at which the lines specified in the preceding operand are to be inserted. If a line with this number exists, the lines are inserted following the existing line. If this operand is omitted, the lines are added following the last line of the active data set. The MERGE subcommand is used to combine a CRJE data set with the active data set or to copy lines from one place to another within the active data set.

The lines that are inserted, and any subsequent lines in the active data set, are renumbered according to the line increment attribute of the active data set.

Examples

1. m mydata

As a result of this subcommand, all of the lines in MYDATA, a CRJE data set in the library of the user entering the subcommand, are copied into the active area following the last line in the active data set. When the lines become a part of the active data set, they are renumbered, starting after the last existing line number and using the line increment attribute assigned to the data set. For example, if the last line in the active data set is 200 and the line increment attribute is 10, then the added lines will be numbered 210, 220, 230, etc.

2. merge jcl (howard) 0

In this example, JCL is a CRJE data set in the library of the user with the userid HOWARD. When this form of the subcommand is entered, the entire data set called JCL is inserted at the beginning of the active data set. The resulting data set is renumbered in its entirety.

active data set (before)		active data set (after)	
line		line	
10	SOURCE 1	10	JCL 1
20	SOURCE 2	20	JCL 2
30	SOURCE 3	30	JCL 3
		40	JCL 4

110 SOURCE 11
120 SOURCE 12

50 JCL 5
60 JCL 6
70 SOURCE 1
80 SOURCE 2
.
170 SOURCE 11
180 SOURCE 12

3. m * 10,30 110

Following the subcommand, lines 10, 20, and 30 in the active data set are inserted following line 110.

active data set (before)
line

10 SOURCE 1
20 SOURCE 2
30 SOURCE 3
.
.
110 SOURCE 11
120 SOURCE 12

active data set (after)
line

10 SOURCE 1
20 SOURCE 2
30 SOURCE 3
.
.
110 SOURCE 11
120 SOURCE 1
130 SOURCE 2
140 SOURCE 3
150 SOURCE 12

RENUMBER: The RENUMBER subcommand requests that new line numbers be assigned to the lines in the active data set.

Operation	Operand
REN[UMBER]	[linenum [increment]] [10 [10]]

linenum [increment]

Specifies the numbering scheme to be used in renumbering the lines. Numbering begins with the specified line number and continues according to the specified increment. If the increment is not specified, an increment of 10 is used. The increment used becomes the line increment attribute of the data set.

no operand

Specifies that line numbering begin with 10 and proceed in increments of 10.

All of the lines in the active data set are renumbered. If the data set has the SEQ attribute, then the new line numbers are also retained in the last eight positions of the records.

A numbering scheme can be specified even if there are no lines in the active data set. This is a convenient way to change the line increment attribute prior to creating a data set.

Examples

1. ren

When this subcommand is entered, all the lines in the active data set are renumbered, starting with 10 and continuing in increments of 10. Ten becomes the line increment attribute of the data set.

2. renumber 100,50

This subcommand requests that the lines in the active data set be renumbered, starting with 100 and continuing in increments of 50. The line increment attribute of the data set becomes 50.

LIST: The LIST subcommand is used to display a line or a group of lines in the active data set.

Operation	Operand
L[IST]	[linenum [linenum]] [<u>NUM</u> NONUM]

linenum [linenum]

Specifies the line or the range of lines to be listed. If a range is specified, those lines with line numbers the first line specified and the last line number specified are listed. If no line numbers are specified, the entire active data set is listed.

NUM

Specifies that line numbers are to be listed with the data. If the data set has the SEQ attribute, the line number will only be listed once, that being at the beginning of the line.

NONUM

Specifies that line numbers are not to be listed with the data. Positions 73-80 are not printed if the data set has the SEQ attribute.

The user can terminate the listing by following the interrupt procedures defined for his terminal type. If the terminal does not have an interrupt feature, the listing is returned in groups of "n" lines, where "n" is an installation-defined value. The user can terminate the listing between groups of "n" lines. If the terminal has an interrupt feature, the listing proceeds without stopping. The user can interrupt at any time. To complete a listing that has been interrupted, the user must reenter the LIST subcommand specifying the lines he wants listed..

Note 1: You should not use the LIST subcommand with unprintable data such as object modules, for they may contain embedded line control characters. Doing this may cause a TP line error.

Note 2: If the active data set has the NOSEQ attribute (specified by the NONUM operand on the EDIT command), the entire 80 characters of the line will be listed. For the SEQ attribute, only 72 characters will be listed.

Examples

1. l

This subcommand, with no operand, results in a listing of all the lines in the active data set. The line number is listed preceding each line. If the user's terminal does not have an interrupt feature, the user may have to take positive action to receive each group of "n" lines. (See appendixes for the operating procedures of each terminal type.)

2. list 0, 100

This subcommand results in a listing, starting with the first line in the active data set and continuing through line 100.

SCAN: The SCAN subcommand requests a syntax analysis of a statement or group of statements in the active data set. This subcommand is also used to control the automatic syntax scan of lines entered in input mode.

Operation	Operand
SC[AN]	[[linenum [linenum]] [ON OFF]

no operand

Specifies that the entire active data set be scanned for syntax errors.

linenum [linenum]

Specifies the line or the range of lines to be scanned. If a source statement does not end on the last line specified, scanning continues to the end of the statement or until the installation-imposed maximum number of lines is reached.

ON

Specifies that all lines subsequently entered in input mode are to be scanned for syntax errors.

OFF

Specifies that lines subsequently entered in input mode are not to be scanned for syntax errors.

The SCAN subcommand is valid only if the active data set has the PL1 or FORT content attribute and if the installation has included the corresponding syntax checker in the system. When a syntax checker is included in the system, the installation specifies the maximum number of lines that can be passed to the checker at one time. CRJE collects up to the maximum number of lines before passing them to the syntax checker. If a source statement takes more than the maximum number of lines, the partial statement is passed to the checker.

When syntax errors are detected, either as a result of a specific request by the SCAN subcommand or during the automatic scanning of lines entered in input mode, the user receives a diagnostic message indicating the number of the line that contains the error. If the error is detected in input mode using the automatic scan, the user can either ignore the error by entering a null line or he can correct the error by using the Implicit or CHANGE subcommands. Only lines contained in the group of lines just syntax checked may be corrected. When the error has been corrected, a null line of input causes the group of lines to be rechecked, after which input mode is reentered.

If an error is detected as the result of a specific request for existing lines, correction is left to the user, who can correct the error using any of the EDIT subcommands. There is no automatic rechecking of corrected statements. They can be rechecked by reissuing the SCAN subcommand or by using the automatic scan while entering the corrections.

Examples

1. `sc 50`

This subcommand requests that the statement beginning on line 50 of the active data set be scanned for syntax errors. If the statement is continued on subsequent lines, scanning continues to the end of

the statement or until the maximum number of lines has been collected for the syntax checker. The automatic scan is not affected.

2. scan on

This subcommand requests automatic syntax scanning of lines subsequently entered in input mode. The automatic scan does not apply to lines entered with the Implicit subcommand.

SAVE: The SAVE subcommand is used to store the active data set in the user's permanent library.

Operation	Operand
S[AVE]	[dsname] [K[EY](key)]

dsname

Specifies the simple name under which the active data set is to be stored. The name consists of 1-8 alphameric characters, the first of which must be alphabetic. If no dsname is specified, the dsname specified in the EDIT command is used. If the EDIT command specified an OS data set, then the dsname operand is required. If not supplied, the system will prompt the user for a name.

K[EY](key)

Specifies that the data set be assigned the specified key for protection from unauthorized access. If this operand is specified, another user must enter the key as well as the dsname and userid to examine the data set. If a key is not specified, another user has access to the data set by supplying only the dsname and userid. Key can be any three-character identifier.

The active data set is saved in the library of the user entering the command. The data set is stored with the attributes specified on the EDIT command.

If the active data set is a copy of a CRJE data set that has a protection key, that key will still be associated with the data set when it is saved unless it is overridden by another key, or a dsname is specified on the SAVE subcommand. Specifying a dsname but no key when saving a data set deletes any existing protection key.

If a data set has just been created, if a data set in another user's library was specified on EDIT, or if a dsname is specified differently from the dsname on EDIT, then CRJE checks in the user's library for an existing data set with the same name. If a duplicate is found, the user is prompted for a new dsname. If the user responds with a null line, the active data set replaces the data set in the user's library.

If the active data set is a copy of a CRJE data set in the library of the user entering the subcommand and a different dsname is not specified on SAVE, then the active data set replaces the CRJE data set with the same name.

If there is not enough space available in the user's library, the library is automatically condensed in order to free any unused space. If sufficient space is not available after condensation, the user is prompted for the name of a CRJE data set to be deleted from his library. If the user responds with a null line, processing of the SAVE subcommand is terminated and the active data set is not saved. If the user responds with the dsname of a CRJE data set in his library, that data set is deleted and the prompt is repeated. When the user has finished

deleting data sets, he responds to the prompt with a null line. His library is then condensed, and there is another attempt to save the active data set.

If records are being saved into an 80-character record user library instead of an 88-character record library, the last eight positions (73-80) of each record are checked to determine if the line number or blanks are located there. If so, the record(s) is saved, provided that the library is not full. If data is there instead of blanks or the line number, the system overlays the data with the line number and sends the message DATA LOST IN TRUNCATION to the user. If the overlaid data needs to be saved, reenter the record on two lines, if necessary, restricting each line to a maximum of 72 characters.

Examples

Since the action taken for a SAVE subcommand is also dependent upon the EDIT command that was issued, the examples show combinations of EDIT and SAVE.

1. edit data,new
save

In this example, the new data set with the dsname DATA is saved in the user's library. If his library already contains a data set with the name DATA, the user will be prompted for another dsname.

2. edit data
s

In this case, the active data set is a copy of a CRJE data set in the user's library. When the user enters the SAVE subcommand, the active data set replaces the original copy in the user's library. When the user enters the SAVE subcommand, the active data set replaces the original copy in the user's library.

3. edit data (jones)
s

Here, the active data set is a copy of the CRJE data set, DATA, in the library of the user with the userid JONES. It is saved with the dsname DATA in the library of the user entering the SAVE subcommand. The user will be prompted for a new dsname if his library already contains a data set named DATA.

4. edit 'howard.results',nonum
s results k(how)

In this example, the active data set is a copy of the OS data set, HOWARD.RESULTS. The active data set is saved in the user's library, and is named RESULTS. HOW is the protection key assigned to the data set. If a data set named RESULTS already exists in the user's library, he is prompted for another dsname.

5. edit ex1
save ex2

The active data set is a copy of EX1, a CRJE data set in the user's library. The user specifies a new dsname, EX2, when he saves the data set, so that there will be a data set named EX1 and a data set named EX2 in his library. If the name EX2 already exists, the user will be prompted for another dsname.

6. edit job1 (brown,xxx)
save brn job

The active data set is a copy of the protected data set JOB1 in the library of the user with the userid BROWN. It is saved with the

dsname BRNJOB in the library of the user entering the SAVE subcommand. Whereas JOB1 was protected with the key XXX, the data set BRNJOB has no protection key. If a dsname has not been specified on SAVE, BRNJOB would have the protection key XXX.

END: The END subcommand terminates creating and editing operations by deleting the active data set.

Operation	Operand
END	(no operand)

Since the active data set is deleted as a result of the END subcommand, the user must have previously issued the SAVE subcommand if he wants a permanent copy of the data set.

With the END subcommand, the user returns to command mode, in which EDIT subcommands are not valid. In order to resume editing or creating, the user must reissue the EDIT command.

Example

1. end

The active data set is deleted, and the user cannot do any more creating or editing until an EDIT command is entered to establish another active data set.

DELETE

The DELETE command is used to scratch an OS data set or to remove a CRJE data set from the user's own library.

Operation	Operand
DELETE	dsname

dsname

Specifies the name of the data set to be deleted. Depending upon whether the data set is a CRJE data set or an OS data set, this operand takes one of two forms:

dsname: Specifies the simple name of a CRJE data set. The data set must reside in the library of the user entering the command.

'dsname[(member name)]': Specifies the fully qualified data set name of an OS data set. The data set must be sequential or partitioned (a member of a PDS can be specified), cataloged, and resident on a single volume of a direct-access storage device. The first qualifier of the dsname must be the userid of the user entering the command. The entire operand must be enclosed in single quotes.

The DELETE command is only valid in command mode. It should not be confused with the DELETE subcommand, which is used in edit mode to delete lines of an active data set.

The space occupied by a deleted CRJE data set becomes available only after the library has been condensed to make space available to save a data set.

The catalog entry of an OS data set is deleted if the data set is sequential or if it is partitioned and no member name is specified in the dsname operand. The catalog entry is not deleted if a member of a partitioned data set is specified.

The user should delete OS data sets created by jobs he has submitted as soon as he has finished with them. By so doing, he avoids tying up storage space at the central installation.

JOB PROCESSING COMMANDS

SUBMIT

SUBMIT can be entered in either command mode or edit mode. It is used to enter one or more jobs into the OS job input stream for batch processing.

Operation	Operand
SUB[MIT]	{ dsname } ... { * }

Note: The abbreviated form, SUB, is only valid in edit mode.

dsname

Specifies either a DSLIST data set or a data set, the contents of which are to be incorporated into the input stream. This must be the name of a CRJE data set and must have one of the following formats:

dsname: specifies the simple name of a data set in the library of the user entering the command.

dsname (userid): specifies an unprotected data set in the library of the user identified by userid.

dsname (userid key): specifies a protected data set in the library of the user identified by userid. Key specifies the protection key assigned to the data set.

*

Specifies that the contents of the active data set are to be incorporated into the input stream. This operand is valid only in edit mode. The active data set may have the DSLIST attribute.

Up to ten operands can be specified. The data sets are entered into the input stream in the order they are specified. If a data set with the DSLIST attribute is specified, the contents of all the data sets named in the DSLIST are submitted.

The JCL statements are checked by an installation exit routine, if one is provided. The user may receive messages generated by the installation exit routine. The exit routine may also request that the submission be aborted. Such a request does not affect any records encountered prior to the abort request.

The contents of the specified data sets are passed to the OS job management routines for batch processing. The user can continue with his session while the job is executed in the background. When the job

is completed, the user will receive a message indicating normal or abnormal termination.

The user may inquire about the status of jobs he has submitted by entering the STATUS command. He may cancel a job he has submitted by entering the CANCEL command.

Note 1: CRJE does not support the use of instream procedures under MFT.

Note 2: The user must guard against using JOBFAIL as the name of a remotely submitted job because the operating system assigns that name to all jobs having JCL errors on the JOB statement.

Examples

1. submit myjob

The CRJE data set, MYJOB, contains a complete job stream, including all necessary job control statements, program source statements, and data. MYJOB is a data set in the library of the user entering the command. This command could be entered in command mode or edit mode since the active data set is not specified.

2. submit jcl(jones), myprog, *

In this example, SUBMIT is used in edit mode, since the active data set is part of the input stream. The CRJE data set, JCL, is in the library of the user with the userid JONES, whereas MYPROG is in the library of the user entering the command. The contents of all three data sets, in the order specified, make up the job stream.

3. sub *

For this example, assume the active data set has the DSLIST content attribute and contains the following lines:

```
line
10 JCL1
20 RTN1
30 RTN2
40 RTN3
50 JCL2
60 DATA1 (HOWARD)
```

Each line is the name of a CRJE data set. The contents of all the data sets make up the job stream. This form of SUBMIT is valid only in edit mode.

CANCEL

CANCEL is used to remove a job from the CRJE system. It can be entered in either command mode or edit mode.

Operation	Operand
CA[NCEL]	jobname

Note: The abbreviated form, CA, is only valid in edit mode.

jobname

Specifies the name of the job that is to be removed. The jobname is the same name specified in the name field of the JCL JOB statement.

The user can cancel any job that he has previously submitted. The job can be canceled while it is still on the operating system job queue scheduled for initiation, while it is currently being executed, or after it has terminated either normally or abnormally. When the job is actually removed from the system depends upon its status at the time the CANCEL is entered. If the job has not yet been placed on the job queue, the job is not canceled, but at completion, all of the job's output that is in the CRJE SYSOUT class is deleted. If the job is on the job queue, it is flagged to be removed from the system when it is initiated. If the job is being executed, the CANCEL is effective immediately. If the job is completed, all of that job's output that is in the CRJE SYSOUT class is deleted.

When the job is actually removed from the system, the user receives a message indicating that the job is deleted. Not until that message is received can the user submit another job with the same jobname.

Example

1. cancel test1

In this example, TEST1 is the name of a job the user has submitted. When deletion of the job is accomplished, the user will receive the message, JOB CANCELLED TEST1.

OUTPUT

The OUTPUT command is used to request CRJE SYSOUT output of a conversationally submitted job.

Operation	Operand
OUTPUT	jobname [MSG]

jobname

Identifies the job whose output is requested. If this is the only operand specified, only the data sets in the CRJE SYSOUT class are returned.

SMSG

Specifies that job management messages are to be returned in addition to output data sets.

Output that can be requested with the OUTPUT command consists of data sets and OS job management messages that are in the CRJE SYSOUT class. Data sets are directed to this class by specifying, in the JCL Data Definition (DD) statement, the SYSOUT class assigned to CRJE by the central installation. Messages are routed to the CRJE SYSOUT class if the MSGCLASS parameter of the JOB statement specified the CRJE SYSOUT class.

If the logical record size of an output data set exceeds 120, the records are truncated to 120 characters.

The data sets are returned in the same order that the DD statements describing them appeared in the input stream. After a SYSOUT data set has been transmitted in its entirety, the data set is deleted. The job management messages are deleted either after all of them have been transmitted or after all the output data sets of the job have been transmitted.

The terminal user can discontinue the transmission of output by following the interrupt procedures defined for his terminal type. The output is returned according to whether or not his terminal has an interrupt feature. If the terminal does not have an interrupt feature, the output is returned in groups of "n" lines, where "n" is an installation-defined value. The user can discontinue receiving output between groups of "n" lines. If the terminal has an interrupt feature, the listing proceeds without stopping. The user can interrupt at any time.

The CONTINUE command can be used to resume output listing that has been discontinued. The CONTINUE command must be entered during the same session in which the interrupt occurred, and must be entered before another OUTPUT command is entered by the same user. Otherwise, the OUTPUT command can be used to list the remainder of the job's output in the CRJE SYSOUT class.

Note: If control characters are present in the data set requested with the OUTPUT command, they will be removed before the actual output is transmitted to the terminal. You should not use the OUTPUT command with unprintable data such as object modules, for they may contain embedded line control characters. Doing this may cause a TP line error.

Examples

1. output test1

As a result of this command, the contents of all the CRJE SYSOUT data sets produced by the job TEST1 are printed at the user's terminal. If the terminal does not have an interrupt feature, the user may have to take positive action to receive each group of "n" lines. If by chance the user previously requested the output but had interrupted the transmission of it to his terminal, this command results in the delivery of those data sets that are left in the CRJE SYSOUT class.

2. output test1, msg

This command requests job management messages as well as CRJE SYSOUT data sets. The job management messages are available only if the MSGCLASS parameter of the JOB statement for TEST1 specified the CRJE SYSOUT class.

CONTINUE

Used in conjunction with the OUTPUT command, the CONTINUE command requests that the output that was previously interrupted be resumed.

Operation Operand

Operation	Operand
CONTINUE	H[ERE] B[EGIN] N[EXT]

H[ERE]

Specifies that transmission of output is to be resumed with the first record of the SYSOUT block that was being written when the output was discontinued. If records are blocked, there is a possibility of duplicate records, since transmission is resumed at the beginning of a block rather than with a specific record.

B[EGIN]

Specifies that, if a data set was being transmitted when output was discontinued, output is to be resumed starting with the beginning of the data set. If system messages were being transmitted, output is resumed with the block of messages that was interrupted.

N[EXT]

Specifies that a data set that was being transmitted when output was interrupted be deleted and output be resumed with the next SYSOUT data set or block of messages.

The CONTINUE command is only valid after the user has discontinued receiving CRJE SYSOUT output that he previously requested with the OUTPUT command. This OUTPUT command must have been entered during the user's current session and must be the last OUTPUT command entered before the entry of the CONTINUE command. If a new session is in progress, the user must reenter an OUTPUT command to receive the remaining output of a job. The OUTPUT command can also be entered to continue receiving output that was interrupted in the same session. The results would be the same as those following a CONTINUE BEGIN command, unless the OUTPUT command requests system messages, in which case all system messages are retransmitted instead of only the current block of messages.

Examples

1. continue

Assuming that the user interrupted the transmission of the output requested by the last OUTPUT command issued in this session, this CONTINUE command, with no operands, requests that the output be resumed at the point it was interrupted. If the user had requested job management messages by specifying SMSG on the OUTPUT command then he will continue to receive the messages after the CONTINUE command.

2. continue n

As a result of this command, a data set that was being transmitted when the user interrupted the output requested by the last OUTPUT command is deleted. Transmission of output is resumed with the next CRJE SYSOUT data set, or possibly the next block of messages if SMSG was specified on the OUTPUT command.

STATUS INFORMATION COMMANDS

STATUS

The STATUS command is used to request information about jobs the user has submitted.

Operation	Operand
STATUS	[jobname]

jobname

Requests the status of the job identified by jobname.

no operand

Requests a list of all the jobs submitted by the user, and the status of each job.

Only the user who submitted the job can request the status of it. A job is considered to be in the CRJE system as long as the CRJE SYSOUT class contains output for the job.

The status information returned indicates whether a job is scheduled for execution, is currently being executed, is completed, or is not in the system. If a job is scheduled for execution, its position on the job queue is provided.

Examples

1. status

After this command, with no operand, the user receives a message for each job he has submitted that is still in the system. Each message (IHK385) gives the jobname and the status of the job.

2. status test1

This command requests the status of the job with the jobname TEST1. The message returned, IHK385, indicates whether the job is scheduled, executing, or completed.

LISTDS

The LISTDS command is used to request information about a CRJE data set in the user's library.

Operation	Operand
LISTDS	dsname [S[TATUS]] [H[ISTORY]]

dsname

Specifies the CRJE data set about which information is requested. The data set must be in the library of the user entering the command. Accordingly, dsname is the simple name assigned to that data set. This operand is required and must be the first operand specified. If it is the only operand, the information returned will be the content attribute of the data set and an indication of whether line numbers are maintained in positions 73-80 of each record.

S[STATUS]

Requests the size of the data set and an indication of whether the data set is assigned a protection key.

H[HISTORY]

Requests the following information about the data set: creation date (first time saved), date last modified (last time saved), and the number of times specified since created. An access of data set occurs when the data set is specified in one of the following commands and subcommands: EDIT with the OLD operand, SUBMIT, EXEC, and MERGE.

The information requested by a STATUS or HISTORY operand is returned in addition to the content attribute and the sequence attribute (i.e., line numbers in positions 73-80).

Since the data set ACTIVE does not have attributes, that data set name will be followed by asterisks instead of the usual status and/or history information.

The LISTLIB command can be used to obtain the names of all the CRJE data sets in the user's library and information about each.

Note: The content and key attributes displayed for formerly non-CRJE data sets are unreliable unless established by the user. The content attribute is set as specified in the EDIT command. The key attribute may be set by specifying the desired key in the KEY operand of the initial SAVE command. The key may also be cleared by specifying only the desired dsname as an operand of the initial SAVE command rather than waiting to be prompted for a dsname. The key can be set or cleared on any subsequent SAVE command.

Examples

1. `listds myprog`

After this command is entered, the user will receive message IHK320, which will give the content and sequence attribute of the data set MYPROG. The information will be given under the headings, TYPE (the content attribute) and LNO (YES for SEQ attribute; NO for NOSEQ attribute).

2. `listds myprog, s,h`

As a result of this command, the message IHK320 will include, in addition to the TYPE and LNO headings, the following headings: BLOCKS (the number of blocks of 10 lines), KEY (YES for protected; NO for unprotected), CREATED (date first saved), LAST CHG (date last saved), and XREAD (the number of times the data set has been specified).

LISTLIB

The LISTLIB command is used to list the names of all the CRJE data sets in the user's library with information about each.

Operation	Operand
LISTLIB	[S[STATUS]][H[HISTORY]]

no operand

Requests the name, content attribute, and sequence attribute of every CRJE data set in the library of the user entering the command.

S[TATUS]

Requests, in addition to the names and attributes of the data sets in the user's library, the size of each data set and whether it is protected.

H[ISTORY]

Requests, in addition to the names and attributes of the data sets in the user's library, the following information: creation date (first time saved), date last modified (last time saved), and the number of times specified since created. An access of a data set occurs whenever the data set is specified in one of the following commands and subcommands: EDIT with the OLD operand, SUBMIT, EXEC, and MERGE.

Since the data set ACTIVE does not have attributes, that data set name will be followed by asterisks instead of the usual status and/or history information.

Information about a particular data set in the user's library can be obtained with the LISTDS command.

Note 1: Should the system abnormally close down while you are receiving the contents of your user library via LISTLIB, message IHK389 SESSION TERMINATED - ACTIVE AREA SAVED - DSNNAME=ACTIVE is sent to you after CRJE warmstarts. This saved data set contains the output you had received up to the point of abnormal closedown.

Note 2: The content and key attributes displayed for formerly non-CRJE data sets are unreliable unless established by the user. The content attribute is set as specified in the EDIT command. The key attribute may be set by specifying the desired key in the KEY operand of the initial SAVE command. The key may also be cleared by specifying only the desired dsname as an operand of the initial SAVE command rather than waiting to be prompted for a dsname. The key can be set or cleared on any subsequent SAVE command.

Example

1. listlib s,h

In response to this command, the user will receive message IHK320, which consists of headings for columns containing the requested information. The headings and their meanings are as follows: DSNNAME, the simple name of the data set; TYPE, the content attribute of the data set; LNO, YES for the SEQ sequence attribute or NO for the NOSEQ attribute; BLOCKS, the number of 10-line blocks in the data set; KEY, YES if protected or NO if unprotected; CREATED, the date first saved; LAST CHG, the date last saved; and XREAD, the number of times accessed since created.

MESSAGE COMMANDS

SEND

SEND is used to send a message to the central operator or to another terminal user. It can be entered in either command mode or edit mode.

Operation	Operand
SEND	'text' [U[SER](userid) [N[OW] L[OGON]]] O[PERATOR](integer)

'text'

Specifies the text of the message to be sent. The message text may include as many as 40 characters. The single quotes enclosing the text may be omitted if there are no commas, blanks, or parentheses within the text. If single quotes are used to delimit the text, any single quotes to be included in the text (e.g., apostrophes) must be paired. The pair will count as one character.

This must be the first operand entered. If this is the only operand specified, the message is sent to the central operator. If the system supports multiple consoles, the message is sent to the console selected by the installation when the system was generated.

U[SER](userid)

Indicates that the message is to be sent to the terminal user with the specified userid.

N[OW]

Specifies that a message directed to a terminal user be delivered only if the user is currently logged on the system.

L[OGON]

Specifies that, if a terminal user is not currently active, the message is to be delivered to him when he next logs on the system. If this operand is not specified and the user to receive the message is inactive, the message is ignored and the sender is notified.

O[PERATOR](integer)

Indicates that the message is to be sent to the central console for which the specified integer is the routing code. This operand is significant only for systems supporting multiple central consoles. The integer specifying the routing code can range from 1 to 16 inclusive. If zero (0) is used, however, it has the same effect as 1; i.e., the message is sent to the master console. NOW and LOGON are ignored when used with OPERATOR.

A message directed to the central operator is queued for immediate delivery, as is a message directed to a terminal user who is active. The sender may assume this is the case when he receives an underscore prompt for his next command.

If, however, a user to whom the message is directed is inactive, the sender is notified of the action taken, which is determined by the NOW and LOGON operands. The sender is also notified of any unusual situations that may have occurred while manipulating the message, such as a disk error.

Examples

1. `send 'is volume 666666 mounted'`

As a result of this command, the central operator will receive the message, CRJE MSG FR JONES IS VOLUME 666666 MOUNTED, assuming JONES is the userid of the user entering the SEND command.

2. `send hello u(brown)`

If the user with the userid BROWN is active, he will receive the message, CRJE MSG FR JONES HELLO, assuming JONES is the sender. If BROWN is not active, JONES will receive the message, USER INACTIVE, MSG NOT SAVED.

3. `send 'don't use my library today', u(brown), 1`

The user, BROWN, will receive the message, CRJE MSG FR JONES DON'T USE MY LIBRARY TODAY. If BROWN is active, then JONES will simply receive a prompt for his next command and BROWN will receive the message prior to his next underscore prompt. If BROWN is currently inactive, then JONES receives the message, MSG SAVED FOR LOGON, and BROWN will get the message when he logs on.

LISTBC

The LISTBC command is used to request a copy of the broadcast messages.

Operation	Operand
LISTBC	(no operand)

The broadcast messages contain information of general interest to all users. The messages are created and maintained by the central operator.

Each broadcast message is listed with an identifier, which determines the order in which the messages are kept.

If his terminal doesn't have an interrupt feature, the user may have to take positive action to receive each group of "n" lines of messages. If the user interrupts the output while receiving the broadcast messages, he must reenter the LISTBC command to receive them again.

Example

1. `listbc`

This command requests a list of the current broadcast messages.

SPECIAL COMMANDS

TABSET

TABSET is used to indicate logical tab settings for input from a terminal and physical tab settings for output to a terminal. It can be entered in either command or edit mode.

Operation	Operand
TAB[SET	[num...] [IN[PUT OUT[PUT]
	OFF

Note: The abbreviated form, TAB, is only valid in edit mode.

num...

Specifies the tab settings for either input or output. If this is the only operand specified, it applies to input. A maximum of 10 positions may be specified. If more than 10 are specified, the first ten are used, and the user is notified of the action taken. Zero and one have no meaning as tab settings. They are ignored and will not count in the ten allowed tab specifications. (See the third example.)

Input tabs are logical tab settings. Each number specified indicates where each field is to begin from the origin of the text, excluding the line number prompt. To CRJE the first position where a field can begin--logical position zero--is the same as the tenth physical position at the terminal because of the 10-space requirement for the line number prompt. (Remember that CRJE starts counting at zero, not at one. If you will look at your terminal margin, you will notice that the first number is zero, not one.) When an input tab is specified, it is interpreted in the stored input record as one character with blanks inserted up to the next specified tab character. This tab character is inserted whenever the user depresses the TAB key at his terminal. The maximum permissible input tab setting is 80.

Output tabs indicate physical tab settings. Each number specified indicates where each field is to begin from the left margin of the terminal keyboard, which is considered position zero. The maximum permissible output tab setting is 120.

OFF

Specifies that tab settings are no longer to be used for either input or output. If this is the only operand, it applies only to input tabs.

IN[PUT]

Specifies that this command applies to all lines entered from the terminal.

OUT[PUT]

Specifies that this command applies to lines sent to the terminal.

The user is responsible for setting and resetting the physical tabs on his terminal. Any input lines required in setting or resetting tabs must be canceled after the tabs have been set. (See the appropriate appendix for canceling lines on your particular terminal type.)

If the number of physical tabs entered in a line exceeds the number of logical tab settings, each of the additional physical tabs is translated into a single blank.

The ten positions required for the line sequence number (or line number prompt) should be allowed for when tabs are physically set at the terminal. (The line number prompt and the line sequence number are the same. When data is being entered, the number of the line being entered is considered the line number prompt; afterwards, when the data is being listed, modified, etc., the number preceding the line of input is

considered the line sequence number.) The user should set his physical tabs for 10 more than the corresponding logical tabs specified in the command. For example, a tab may be specified on the command for logical position 20, but it should be set physically at position 30 to allow for the line sequence number. (See the first two examples.)

When listing a data set, line sequence numbers need not be considered if the NONUM operand on the LIST subcommand is specified. With NONUM specified, the line sequence numbers are not printed and, therefore, the logical tab settings may be specified so that they directly correspond to the physical tab settings. Otherwise, when a data set is listed, there will be a 10-space offset to the right of the physical tab setting to allow for the 10 spaces required for the line sequence number.

If TABSET is used with the Implicit subcommand, the format of the output will appear differently from the way it was specified. CRJE subtracts "n" characters from the specified setting and uses the result as the actual tab setting, where "n" equals the sum of the characters -- line number and single-character delimiter -- that precede the text in the Implicit subcommand.

Examples

```
1. tabset 30 50 70
   tabset 30 50 70 out
   edit mtab new
```

```
      10      1111      2222      3333
      20      4444      5555      6666
      30      7777      8888      9999
      40      0000      1111      2222
```

list

```
      10              1111      2222      3333
      20              4444      5555      6666
      30              7777      8888      9999
      40              0000      1111      2222
```

list nonum .

```
      1111      2222      3333
      4444      5555      6666
      7777      8888      9999
      0000      1111      2222
```

The data set MTAB is created with input tabs set for logical positions 30, 50, and 70. After receiving the line number prompt, the TAB key is depressed, causing the carriage to tab to position 30 rather than your having to depress the space bar 20 times to that position. (Remember that you would have to space 20 times instead of 30 times because the first 10 spaces are used for the line number prompt.) Data is entered in the first field; the TAB key is then depressed again, causing the carriage to tab to position 50. The same action is repeated for position 70. Input tabs were set in this data set to save time when entering data.

When MTAB is listed, with output tabs set for physical positions 30, 50, and 70 notice what happens: the line sequence number is printed, and the carriage moves (via TAB) to the 30th physical position. However, physical position 30 is logical position 20 (because of the 10-space offset for the line number prompt), and the data was entered at logical position 30. So, the carriage will space 10 more positions to logical position 30 (which is physical position 40), where the data is then listed. The carriage moves to logical position 50, spaces 10 more positions to physical position 60, and the data is again listed. This action is repeated for the third tab specification. This ten-space offset can be avoided in one of two ways: 1) by specifying the logical

tabs for 10 less than the physical tabs, or 2) by listing the data set and specifying the NONUM operand of the LIST subcommand, which is illustrated in this example. By listing MTAB with NONUM specified, the data in each record begins in physical positions 30, 50, and 70, respectively. Notice, then, that when a data set is listed in this manner, it appears physically in the same way as it did when it was created. (The physical appearance would be the same, also, if the input tabs were specified for 10 less than the output tabs.)

2. tabset 6

edit fortab new fortg s

```

10      prin=24.00
20      do 10 i=1, 349
30      10 prin=prin*1.03
40      write (6,15) prin
50      stop
60      15 format ('your answer is', f20.1)
70      end

```

save

This example illustrates the major advantage of input tabs: saving time when data is being entered. Instead of the user's having to depress the space bar six times before entering each FORTRAN statement, he merely depresses the TAB key once and enters each statement (excluding those with statement numbers). Although the tab field is specified for six, the physical tab is set at position 16. This is done for two reasons: 1) CRJE starts counting at zero, not one; and 2) 10 must be added to the desired setting to allow for the line number prompt, which requires ten spaces (otherwise, if a tab field were physically set for six, the tab would not be beyond the line number prompt and would, therefore, be of no use).

3. tabset 0 1 10 20 30 40 50 60 70 80 90 100 110 out

Entering this command will cause two terminal messages to be generated: IHK302 INVALID OPERAND and IHK334 EXCESSIVE OPERANDS. IHK302 will be generated twice: once for 0, and once for 1: both specifications are invalid tab settings and are therefore ignored by CRJE. Because 0 and 1 are ignored, CRJE considers 10-110 as valid tab settings. However, IHK334 will be sent because only ten tab settings are allowed (110 is the eleventh tab setting). Had INPUT been specified instead of OUTPUT, another message, IHK381 TABS EXCEED ALLOWED LIMIT, would have been sent to the user, for 80 is the maximum permissible tab setting for input tabs.

4. tab off

Entering this command clears the tabs that have been specified on any previous TABSET commands. Since neither INPUT nor OUTPUT was specified, input tabs will be cleared by default.

EXEC

The EXEC command is used to execute a series of commands contained in a CRJE data set with the CLIST content attribute.

Operation	Operand
EXEC	dsname [L [IST] [NOL [IST]]

dsname

Specifies the data set that contains the sequence of commands to be executed. This must be a CRJE data set with the CLIST attribute. It can have one of the following formats:

dsname: specifies the simple name of a data set in the library of the user entering the command.

dsname (userid): specifies an unprotected data set in the library of the user identified by userid.

dsname (userid key): specifies a protected data set in the library of the user identified by userid. Key specifies the protection key assigned to the data set.

L[IST]

Specifies that each command in the command list is to be printed at the terminal prior to execution. If this operand is omitted, the commands will not be listed.

NOL[IST]

Specifies that the commands in the command list are not to be printed at the terminal.

The CLIST data set may contain commands and subcommands with the exception of LOGON, LOGOFF, and EXEC. The commands and subcommands are executed as if they were entered from the terminal. If a command or subcommand requires input from the terminal (for example, EDIT and INPUT), the normal messages or line number prompts are received at the terminal to request the input.

The CLIST may be increased dynamically by including in the CLIST an EDIT of the CLIST data set and an INPUT subcommand, and entering input from the terminal. Executing of the CLIST continues as long as the number of blocks in the increased CLIST does not exceed the original number of blocks in the CLIST (10 records equal one block). If the original number is exceeded, execution of the CLIST terminates and the message, IHK345 ORIGINAL NO. OF BLOCKS IN CLIST EXCEEDED, is sent. (See the second example of the EXEC command.)

If an error requires the user to reenter the command, execution of the CLIST is terminated. Execution of a CLIST is also terminated if a LOGON, LOGOFF, or EXEC command is encountered in the command list.

If his terminal does not have an interrupt feature, the user may have to take positive action to get each group of "n" lines of output being sent as a result of a command in the CLIST. If the user interrupts output, the first command he enters following the break determines whether execution of the CLIST is terminated or resumed. To interrupt output and resume execution with the next command or subcommand in the CLIST, the user enters END. (This use of END does not terminate edit mode.) To interrupt output and terminate CLIST execution, the user enters the command or subcommand he wants processed next. To resume receiving output, the user enters a null line.

Note: If (1) the automatic break (interrupt) feature is used to terminate CLIST execution and (2) a CLIST command is the last line listed at the terminal when execution ends, then the command itself is not executed.

Examples

1. exec process

The command list PROCESS will be executed. The commands in the list will not be printed at the terminal.

2. exec excds list

The command list EXCDS will be executed, and the commands will be printed at the terminal. The commands in EXCDS are:

```
10 EDIT EXCDS
20 INPUT
30 SAVE
40 END
```

If, when INPUT executes, three more lines are entered from the terminal, e.g.,

```
50 EDIT EXCDS
60 LIST
70 END
```

after line 40(END) executes, lines 50, 60, and 70 (EDIT, LIST, and END respectively) will then be executed. As many as three more records could be added to this CLIST without terminating its execution, for EXCDS is composed of one block (10 records equal one block). If an eleventh record were added, execution would terminate and message IHK345 would be sent to the terminal.

Message codes do not appear with messages that are sent to a remote terminal unless the user requests them when he logs on the system. Therefore, the CRJE messages are listed according to subject as well as presented in numerical order according to the message code number. The subject listing includes the message codes, which serve as a guide to the numerical listing, where the message is explained and the system action and suggested user action are given.

PL/I and FORTRAN syntax error messages are presented in Appendixes D and E, respectively. Messages originated by terminal users or the central operator do not have message codes. This includes the broadcast messages. The message text in these cases depends upon the text specified by the sender of the message.

In this section, the term command refers to either a command or a subcommand unless specifically noted.

SUBJECT LISTING

LOGON/LOGOFF

IHK301 LOGONS SUPPRESSED
 IHK303 LOGON REJECTED BY INSTALLATION
 IHK308 LOGON REQUIRED
 IHK327 ***CRJE LINE nnn DATE yy.ddd TIME xx:xx***
 IHK350 ENTER USERID
 IHK351 ENTER PASSWORD *****
 IHK352 INVALID USERID OR USERID IN USE, LOGON TERMINATED
 IHK353 INVALID PASSWORD, LOGON TERMINATED
 IHK354 LOGOFF time SESSION TIME xxxx MIN.
 IHK355 USERID IN USE, ENTER USERID

COMMAND SYNTAX ERRORS

General

IHK302 INVALID OPERAND [operand [line number]]
 IHK305 INVALID COMMAND command
 IHK307 INVALID LINE NUMBER line number
 IHK311 QUOTES NOT PAIRED
 IHK313 PARENTHESES NOT BALANCED [dsname[userid][key]linenum]
 IHK314 DELIMITER MISSING
 IHK315 ILLEGAL DELIMITER delimiter
 IHK317 REQ'D OPERAND MISSING (DSNAME
 JOBNAME
 TEXT [1]
 [2]
 LINE NUMBER)
 IHK319 TEXT [1] EXCEEDS 40 CHARACTERS
 [2]
 IHK321 MULTIPLE OR CONTRADICTIONARY USE OF OPERAND operand
 IHK333 INVALID USERID
 IHK334 EXCESSIVE OPERANDS {operand
 [dsname[userid][key]] }

IHK336 INVALID DSNAME dsname[userid][key][linenum]
IHK372 EXTRA COMMAS IGNORED
IHK391 INVALID EDIT SUBCOMMAND command
IHK399 OPERAND NOT IN PARENTHESES operand

EDIT

IHK356 INVALID SOURCE MARGINS
IHK413 DIRECTORY FULL

EXEC

IHK309 INVALID CLIST COMMAND command
IHK345 ORIGINAL NO. OF BLOCKS IN CLIST EXCEEDED
IHK371 INVALID EXEC, DATA SET IS NOT CLIST
IHK380 EXEC TERMINATED

TABSET

IHK381 TABS EXCEED ALLOWED LIMIT
IHK382 TABS NOT IN ASCENDING ORDER

LIBRARY AND OS DATA SET

IHK304 VOLUME NOT MOUNTED volume serial
IHK306 I/O ERROR ON LIB DURING ACTIVE AREA RECOVERY
IHK332 DISK ERROR {dsname}
 {ddname}
IHK338 DATA SET NOT FOUND [dsname][userid][key][linenum]
IHK339 DATA SET NOT PDS OR SEQUENTIAL
IHK347 UNABLE TO OPEN dsname
IHK366 DSNAME MUST BEGIN WITH USERID
IHK370 LIBRARY INOPERATIVE
IHK375 NO DD STATEMENT IN PROCEDURE OR LIBRARY UNAVAILABLE
 [{dsname}] [userid][key][linenum]
 [{ddname}]
IHK384 DATA SET RESIDES ON MULTIPLE VOLUMES
IHK393 EDIT OF DATA SET REQ'D dsname
IHK395 LIBRARY FULL, ENTER DSNAME FOR DELETION
IHK396 ENTER DSNAME FOR DELETION
IHK398 DATA SET NOT ON 2311, 2314, OR 2319 {dsname}
 {ddname}
IHK401 NO MEMBER NAME FOR PDS
IHK402 LINE NUMBER NOT XXXXXXXX OR NUMERICS line number
IHK404 EXPIRATION DATE NOT REACHED - DATA SET NOT DELETED
IHK407 OPERATOR FAILED TO SPECIFY CORRECT PASSWORD
IHK408 INVALID EDIT REQUEST
IHK409 DIRECTORY FULL, ENTER DSNAME FOR DELETION

ACTIVE AREA

IHK316 LINE TRUNCATED, LENGTH EXCEEDS [72] line number
 [80]

IHK340 NO LINES IN RANGE
 IHK341 TEXT NOT FOUND
 IHK343 INVALID INCREMENT - 0
 IHK346 LINE EXISTS, CANNOT INSERT line number
 IHK348 MAX LINE NUMBER EXCEEDED
 IHK349 LINE NOT FOUND line number
 IHK357 ACTIVE AREA OUT OF SPACE
 IHK359 INSERT TERMINATED. NEXT LINE NUMBER IS line number
 IHK361 END OF DATA
 IHK367 INVALID LINE NUMBER RANGE
 IHK389 SESSION TERMINATED-ACTIVE AREA SAVED-DSNAME=ACTIVE
 IHK390 OUT OF SPACE ON LIBRARY DURING ACTIVE AREA RECOVERY
 IHK403 SESSION TERMINATED - ACTIVE AREA LOST
 IHK408 INVALID EDIT REQUEST

SAVE

IHK318 CONDENSE ERROR, DATA SETS MAY BE LOST OR DAMAGED
 IHK369 INVALID KEY {key
 {dsname[userid][key][linenum]}
 IHK392 INVALID SAVE, NO LINES IN DATA SET
 IHK394 DUPLICATE DSNAME, ENTER DSNAME
 IHK395 LIBRARY FULL, ENTER DSNAME FOR DELETION
 IHK396 ENTER DSNAME FOR DELETION
 IHK397 DSNAME INVALID/MISSING, ENTER DSNAME
 IHK410 DATA LOST IN TRUNCATION

JOB PROCESSING (SUBMISSION AND OUTPUT)

IHK310 INVALID NESTING OF DSLIST DATA SETS dsname [userid][key]linenum
 IHK323 INVALID JOB REFERENCE -- JOB NOT SUBMITTED BY USER
 IHK324 JOB NOT COMPLETE
 IHK326 DUPLICATE JOBNAME jobname
 IHK328 MAX JOBS EXCEEDED jobname
 IHK329 JOB TERMINATED {NORMALLY } jobname
 {ABNORMALLY}
 IHK331 JOB CANCELLED [jobname]
 IHK332 DISK ERROR {dsname}
 {ddname}
 IHK338 DATA SET NOT FOUND [dsname][userid][key][linenum]
 IHK344 JOB NOT FOUND jobname
 IHK361 END OF DATA
 IHK365 INVALID JOBNAME jobname
 IHK374 NO DISCONTINUED OUTPUT AVAILABLE
 IHK376 SUBMIT CANCELLED BY INSTALLATION xxxxxxxxx
 IHK378 NO JOBS SUBMITTED

STATUS INFORMATION

IHK320 DSNAME TYPE LNO [BLOCKS KEY] [CREATED LAST CHG XREAD]
 IHK325 NO JOB(S) IN SYSTEM
 IHK344 JOB NOT FOUND jobname
 IHK385 STATUS jobname (SCHED (n)
 EXECUTING
 NORMAL END
 ABNORM END
 NOT QUEUED
 DISK ERROR)
 IHK405 LIBRARY EMPTY

SYNTAX CHECKER ERRORS (FOR STATEMENT ERRORS, SEE APPENDIXES)

IHK360 INVALID SCAN REQUEST, DATA SET NOT PL1 OR FORTRAN
IHK364 LINE line number NOT IN RANGE OF LINES SCANNED
IHK368 TOO MANY CONTINUATION LINES. SCAN line number line number
IHK379 INVALID CORRECTION COMMAND command
IHK386 LEVEL OF SYNTAX CHECKER NOT AVAILABLE
IHK387 SYNTAX CHECKER NOT IN SYSTEM

USER AND SYSTEM MESSAGES

IHK312 NO BROADCAST MSGS
IHK335 MSG SAVED FOR LOGON
IHK337 MSG NOT SAVED
IHK358 MESSAGE LOST
IHK363 USER INACTIVE, MSG NOT SAVED
IHK383 ROUTING CODE NOT 1-16

SYSTEM INFORMATION

IHK300 OUT OF MAIN STORAGE
IHK322 ABNORMAL CRJE CLOSDOWN
IHK330 CRJE CLOSED DOWN
IHK375 NO DD STATEMENT IN PROCEUDRE OR LIBRARY UNAVAILABLE
 [{dsname}] [userid][key][linenum]
 [{ddname}]
IHK389 SESSION TERMINATED-ACTIVE AREA SAVED-DSNAME=ACTIVE
IHK390 OUT OF SPACE ON LIBRARY DURING ACTIVE AREA RECOVERY
IHK400 TP LINE DEACTIVATED BY OPERATOR
IHK403 SESSION TERMINATED - ACTIVE AREA LOST

TERMINAL ERRORS

IHK362 TP LINE ERROR
IHK373 LINE TOO LONG, REENTER LINE
IHK388 TIMEOUT
IHK406 TP LINE ERROR, REENTER LINE

NUMERICAL LISTING

IHK300 OUT OF MAIN STORAGE

Explanation: Temporarily, there is not enough main storage space to process the last command entered.

System Action: Processing of the last command is terminated, and the user is prompted for another command. The central operator is also notified of the condition.

User Action: Reenter the command. If the error recurs, notify the central operator.

IHK301 LOGONS SUPPRESSED

Explanation: The operator has entered a command to prevent, temporarily, any more users from logging on the system.

System Action: The LOGON command is not accepted. The user cannot gain access to the system until the operator releases the LOGON suppression.

User Action: Try the LOGON command again later.

IHK302 INVALID OPERAND operand [[line number]]

Explanation: The operand, which was specified on the last command entered, either 1) is unrecognizable; 2) is an ACCT operand on the LOGON command that contains more than 33 characters of accounting information; , 3) is an ACCT operand with accounting information that was not enclosed in parentheses; 4) is the name of a DSLIST data set that contains an invalid data set name; 5) is invalidly enclosed in parentheses (only the first incorrect operand is specified if more than one operand is enclosed in parentheses); or 6) is 0 or 1 on a TABSET command (both of which are invalid tab settings. If the operand is a DSLIST data set name, the line number of the line that contains the error is also specified. Up to ten characters of the invalid operand are displayed. Invalid parentheses are not displayed.

System Action: No processing is performed for the last command entered, unless it was TABSET. If it was TABSET, the command is processed with the correctly specified operands. In any case, the user is prompted for another command.

User Action: Correct the operand and reenter the entire command, unless it was TABSET. If TABSET was the last command entered, leave the acceptable tab specifications as they are unless you desire to change them. If the operand is a DSLIST data set name, edit the DSLIST data set and correct the error in the specified line number. Then reenter the SUBMIT command. You may also enter a different command following this error.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK303 LOGON REJECTED BY INSTALLATION

Explanation: The LOGON exit routine, which is provided by the installation, has not accepted the LOGON command entered by the user.

System Action: The system waits for the user to reenter the LOGON command. No prompt is issued.

User Action: Determine which information is missing or invalid and log on again. The central operator can tell you what information is required.

IHK304 VOLUME NOT MOUNTED volume serial

Explanation: The disk pack that contains the OS data set specified on the last command is not accessible. The indicated volume serial identifies the disk pack.

System Action: The last command is not processed. The user is prompted for another command.

User Action: Enter another command. You cannot gain access to a data set on the specified volume until CRJE is restarted with that volume mounted. Ask the central operator to mount the disk pack identified by the specified volume serial the next time CRJE is started.

IHK305 INVALID COMMAND command

Explanation: The system does not recognize the indicated verb as a valid command. Up to ten characters of the invalid command are specified.

System Action: No processing is performed, and the user is prompted for another command.

User Action: Determine the correct verb and reenter the command, or enter a new command.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK306 I/O ERROR ON LIB DURING ACTIVE AREA RECOVERY

Explanation: A disk error occurred while trying to save the user's active data set in his library during CRJE warmstart, following an abnormal system closedown or abnormal session termination.

System Action: The active data set is not saved, so there is no CRJE data set named ACTIVE in the user's library. CRJE will continue to handle the user's library, although there is a possibility that the I/O error will recur.

User Action: Proceed with your session. If needed, the active data set that was lost must be reconstructed. If the error recurs, notify the central operator.

IHK307 INVALID LINE NUMBER line number

Explanation: The displayed operand is not a valid line number. It contains either more than eight characters or a nonnumeric character.

System Action: The command containing the invalid line number is not processed. The user is prompted for another command.

User Action: Reenter the last command with the correct line number, or enter a new command.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK308 LOGON REQUIRED

Explanation: The user has entered input without being logged on the system.

System Action: The input that was entered is ignored, and the user is prompted for a command.

User Action: Enter a LOGON command.

IHK309 INVALID CLIST COMMAND command

Explanation: The indicated command has been encountered in the CLIST data set that was being processed as the result of an EXEC command. The command, which is either LOGON, LOGOFF, or EXEC, is not valid in a data set with the CLIST content attribute.

System Action: Processing of the commands in the CLIST is terminated, and the user is prompted for a command to be entered from the terminal. Any commands prior to the invalid command in the CLIST have been processed.

User Action: Continue with your session by entering another command. Before using the CLIST with the invalid command again, edit the CLIST data set and delete the line containing the invalid command.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK310 INVALID NESTING OF DSLIST DATA SETS dsname [userid][key] linenum

Explanation: While processing a DSLIST data set during a job submission, a data set with the DSLIST content attribute was encountered. A data set contained in a DSLIST cannot have the DSLIST attribute. dsname is the name of the nested data set with the DSLIST attribute. If applicable, userid and/or key may be displayed. The line number (linenum) of the nested DSLIST is also printed.

System Action: No more processing is performed for the SUBMIT command that specified the DSLIST containing the nested DSLIST. Any complete jobs collected prior to the error are passed to the operating system for batch processing. (A job is considered complete when another JOB card is encountered.) The user is prompted for another command.

User Action: Continue with your session by entering another command. Before entering the same SUBMIT command, edit the DSLIST data set, replacing the name of the nested DSLIST data set with the names of the data sets in the DSLIST. When resubmitting the job stream, remember not to resubmit any job that was completely collected prior to the error.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK311 QUOTES NOT PAIRED

Explanation: An odd number of single quotation marks has been detected in the last command entered. All single quotes (apostrophes) within delimiting single quotes must be entered as two single quotes.

System Action: No processing is performed for the last command entered, and the user is prompted for another command.

User Action: Correct the error and reenter the last command, or enter a new command.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK312 NO BROADCAST MSGS

Explanation: A listing of the broadcast messages has been requested, but there are no broadcast messages. The messages can be requested by the LISTBC command or by the BC operand (the default options) on the LOGON command.

System Action: The system continues by prompting the user for another command.

User Action: Proceed with your session.

IHK313 PARENTHESES NOT BALANCED dsname[userid][key]linenum

Explanation: In the last command entered, the number of right parentheses does not equal the number of left parentheses. If a DSLIST was being submitted and this error occurred, the dsname in error, along with its associated line number, is also displayed. If a userid and/or key was specified, it is included with the message.

System Action: The last command entered is not processed, and the user is prompted for another command. If the last command was SUBMIT, any complete jobs collected prior to the error are passed to the operating system for batch processing. (A job is considered complete when another JOB card is encountered.)

User Action: Reenter the command, closing all parentheses, or enter a new command. If you are resubmitting a job stream, remember not to resubmit any jobs that were completely collected prior to the error.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK314 DELIMITER MISSING

Explanation: If a CHANGE subcommand was just entered, the second or third delimiter was omitted. Otherwise, the command just entered contained an operand delimited by single quotes, which was not followed by a comma or a blank to separate it from the next operand.

System Action: The command is not processed, and the user is prompted for another command.

User Action: Reenter the command with appropriate delimiters, or enter a new command.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK315 ILLEGAL DELIMITER delimiter

Explanation: The displayed delimiter, which was on the CHANGE or Implicit subcommand just entered, is not a nonnumeric character other than a blank or a comma.

System Action: The CHANGE or Implicit subcommand is not processed, and the user is prompted for another subcommand (since he is in edit mode).

User Action: Reenter CHANGE or Implicit using a different delimiter, or enter another subcommand. If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK316 LINE TRUNCATED, LENGTH EXCEEDS [72] line
number [80]

Explanation: The line just entered in input mode or with the Implicit subcommand contains more than the maximum number of characters allowed. If the data set has the SEQ sequence attribute, 72 characters are allowed; if it has the NOSEQ attribute, 80 are allowed.

System Action: The line is truncated to the appropriate number of characters and is put in the user's active data set. If the line was entered in input mode, the user is prompted with the next line number. If the line was entered with the Implicit subcommand, the user is prompted for his next subcommand.

User Action: If the truncated line is acceptable, continue with your session by entering either the next line if in input mode, or another subcommand if in edit mode. If the truncated line is not acceptable, use EDIT subcommands to delete the line that was entered. If in input mode, you must first enter a null line to enter edit mode so you can issue a subcommand. If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK317 REQ'D OPERAND MISSING {
DSNAME
JOBNAME
TEXT [1]
LINE NUMBER }

Explanation: A required positional operand was omitted on the last command entered. The variable term indicates the operand that was omitted.

1. DSNAME indicates that the data set name was not specified on SUBMIT, LISTDS, EXEC, DELETE, EDIT, or MERGE.
2. JOBNAME indicates that the name of a job was not specified on CANCEL or OUTPUT.
3. TEXT indicates that the message text was omitted on a SEND command. If TEXT1 is displayed, the first character string of the CHANGE subcommand is missing. If TEXT2 is displayed, the second character string of the CHANGE subcommand is missing.
4. LINE NUMBER indicates that the line number was omitted on a CHANGE subcommand.

System Action: No processing is done for the last command entered, and the user is prompted for another command.

User Action: Reenter the command, including the required operand, or enter another command.

IHK318 CONDENSE ERROR, DATA SETS MAY BE LOST OR DAMAGED

Explanation: An I/O error occurred while the user's library was being condensed to make more space available. Some of the CRJE data sets in the library may be missing or may contain errors.

System Action: After recovering as much of the library as possible, CRJE prompts the user for another subcommand.

User Action: You may continue with your session by entering another subcommand, although there is no guarantee that your library is intact. To determine the status of your library, use the LISTLIB command to see if any data sets are missing. Errors within data sets can be determined by listing the data sets. Damaged data sets may be reconstructed. If the error recurs, notify the central operator.

IHK319 TEXT[1]EXCEEDS 40 CHARACTERS
 [2]

Explanation: More than the allowed 40 characters have been specified for the text operand of a SEND command or a CHANGE subcommand. TEXT1 and TEXT2 indicate that the corresponding text operand of the CHANGE subcommand is in error. If neither 1 nor 2 is displayed, the text operand of a SEND command is in error.

System Action: The command or subcommand is not processed, and the user is prompted for another command or subcommand.

User Action: Reenter CHANGE or SEND with less than 40 characters in the text operand, or enter a new command (or subcommand if in edit mode).

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK320 DSNAME TYPE LNO [BLOCKS KEY][CREATED LAST CHG XREAD]

Explanation: These terms are headings for columns of information returned in response to a LISTDS or LISTLIB command. If the STATUS operand was specified on the command, the BLOCKS and KEY headings are included. If the HISTORY operand was specified, the CREATED, LAST CHG, and XREAD headings are included. The categories are as follows:

1. DSNAME - the simple name of the data set about which information is requested. All data sets in the user's library are listed after a LISTLIB command.
2. TYPE - the content attribute of the indicated data set.
3. LNO - YES if the data set has the SEQ sequence attribute (i.e., line numbers in positions 73-80); NO if the data set has the NOSEQ sequence attribute.
4. BLOCKS - the number of groups of 10 lines each that are contained in the data set. The total number of lines in the data set would therefore be between $10(N-1)+1$ and $10N$, where N is the number of blocks.
5. KEY - YES if the data set has a protection key; NO if the data set is unprotected.
6. CREATED - the date the data set was first saved in the user's library. The date is in the form yy.ddd, where yy is the year and ddd is the day of the year.
7. LAST CHG - the date the data set was last changed, that is, last saved in the user's library.
8. XREAD - the number of times the data set has been accessed since it was created. This is the number of times it has been specified on SUBMIT, EDIT with the OLD operand, EXEC, or MERGE.

Note: Attributes assigned to non-CRJE data sets that are listed via LISTLIB or LISTDS are unreliable until the data sets are edited. The type or content attribute is established with the EDIT command. Other data set information is supplied when the data sets are reserved.

System Action: CRJE proceeds by prompting the user for his next command.

User Action: Continue with your session by entering the next command.

IHK321 MULTIPLE OR CONTRADICTIONARY USE OF OPERAND operand

Explanation: In the command just entered, the displayed operand was entered more than once, or it contradicts an operand already specified.

System Action: The command with the duplicate or contradictory operand is not processed, and the user is prompted for another command.

User Action: Reenter the corrected command, or enter a new command. Check the format of the command to determine contradictory operands. Stacked items enclosed in brackets or braces are mutually exclusive.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK322 ABNORMAL CRJE CLOSEDOWN

Explanation: There has been an I/O error in the active area, and CRJE is closing down.

System Action: All users are automatically logged off, and system operation stops. If the central operator specifies the ABNO parameter on the START command when the system is started again, CRJE will try to recover the active data sets in use when the closedown occurred.

User Action: None at that time. You can log on again when the system is restarted. The next time you log on, you will receive a message indicating whether or not your active data set has been recovered and saved in your library under the name ACTIVE.

IHK323 INVALID JOB REFERENCE -- JOB NOT SUBMITTED BY USER

Explanation: The user has attempted to cancel a job that he did not submit or requested its output or status.

System Action: The CANCEL, OUTPUT, or STATUS command is not processed, and the user is prompted for another command.

User Action: Continue with your session by entering another command. You cannot cancel a job that you did not submit; neither can you obtain its output or status.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK324 JOB NOT COMPLETE

Explanation: Output has been requested for a job that has not finished executing.

System Action: The user is prompted for his next command.

User Action: Enter another command. Reenter the OUTPUT command after you have received the message notifying you that your job has completed.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK325 NO JOB(S) IN SYSTEM

Explanation: There is no record of the user's having submitted the job for which he requested status information. If the user requested a list of all his jobs and their status, the message indicates there are no jobs currently in the system that were submitted by the user.

System Action: The user is prompted for another command.

User Action: Continue your session by entering another command.

IHK326 DUPLICATE JOBNAME jobname

Explanation: The input stream specified by the SUBMIT command just entered contains a job that has the same jobname as a job already in the system.

System Action: The input stream collected up to the job with the duplicate jobname is passed to the operating system for batch processing. The rest of the input stream is not processed. The user is prompted for another command.

User Action: Enter another command. To correct the error, edit the data set that contains the JOB card with the duplicate jobname and change the jobname. Then resubmit the portion of the input stream that was not processed. You may also wait for the currently existing job with the same jobname to be removed from the system.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK327 *** CRJE LINE nnn DATE yy.ddd
TIME xx:xx ***

Explanation: The LOGON command just entered has been accepted, and the user is now active. The variables give the following information:

nnn - identification of the communication line by line address.

yy.ddd - the year and day of the year.

xx:xx - the time, according to the system clock, that the session begins.

System Action: After sending any pending messages and, if requested, the broadcast messages, CRJE prompts the user for his next command.

User Action: Enter your next command.

IHK328 MAX JOBS EXCEEDED jobname

Explanation: The user has tried to submit a job, but the installation-specified maximum number of jobs has already been reached. No more jobs will be accepted until some jobs have been removed from the system, either by deletion or by obtaining the output. The displayed jobname identifies the first job not submitted.

System Action: Any jobs preceding the indicated job in the input stream are passed to the operating system for batch processing. The rest of the input stream specified in the SUBMIT command is ignored. The user is prompted for another command.

User Action: Continue your session by entering another command. You can reduce the number of jobs in the system by obtaining the output of, or cancelling, jobs that you submitted that are currently in the system. Otherwise, allow time for other users to remove their jobs before resubmitting the jobs that were not accepted.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK329 JOB TERMINATED { NORMALLY } jobname
 { ABNORMALLY }

Explanation: The job identified by jobname, which the user submitted for batch processing, has completed execution. Output produced by the job is now available. NORMALLY and ABNORMALLY indicate how the job ended.

System Action: The user is prompted for his next command.

User Action: Enter another command. You can request the SYSOUT output of the job by using an OUTPUT command, and the OS data sets created by the job by using EDIT and LIST.

IHK330 CRJE CLOSED DOWN

Explanation: The central operator has requested that CRJE operation stop.

System Action: All users are automatically logged off, and the communication lines are no longer serviced. If a user has an active data set at the time, it is saved in his user library under the name ACTIVE.

User Action: You can log on again when the system is restarted. To resume processing where you left off, edit the data set that was saved in your library under the name ACTIVE.

IHK331 JOB CANCELLED [jobname]

Explanation: The job identified by jobname has been removed from the system as previously requested by the user. If a jobname is not displayed, the cancelled job is the one that was specified on the CANCEL command just entered.

System Action: The user is prompted for another command.

User Action: Continue with your session by entering your next command. You may now submit another job with the jobname of the cancelled job.

IHK332 DISK ERROR {dsname}
 {ddname}

Explanation: An I/O error has occurred on a direct access device other than one in the active area. If the specified name has the form CRJE.LIB.userid, then the error occurred on the library of the user identified by userid, and the library has been marked inoperative. The dsname specified may also be the name of an OS data set the user was attempting to edit. If the error occurs as a result of an EDIT command, the user returns to command mode, and his active data set, if one exists, is deleted.

If the error occurred while reading a SYSOUT data set or building a data set for job submission, then the ddname (i.e., the name field of the DD statement) is displayed.

System Action: No more processing is performed for the last command entered. If the error occurred on a user library, the library is marked inoperative and cannot be used again until the system is restarted. The user is prompted for another command. The central operator is also notified of the error.

User Action: Notify the central operator about the error. You may proceed with other commands if the processing does not require the use of the data set or library on which the error occurred. If the problem does recur, be sure to notify the central operator.

IHK333 INVALID USERID

Explanation: Either the userid specified on a SEND command or the userid identifying the user library of a CRJE data set specified in an EDIT command does not exist.

System Action: The last command entered is not processed. The user is prompted for another command.

User Action: Reenter the command with the correct userid, or enter a new command.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK334 EXCESSIVE OPERANDS [{operand
 {dsname[userid][key]} }]

Explanation: More than the allowable number of operands were entered on the last command. The indicated operand is the first excessive operand.

If the last command entered was SUBMIT, the dsname of the data set in error will be printed. The userid and/or key of the data set is also displayed, if applicable.

System Action: The excessive operands, starting with the one displayed, are not checked for correct syntax, and the command is not processed, unless it was TABSET. If it was TABSET, it is processed using the correctly specified operands. The user is then prompted for his next command.

If SUBMIT was specified, any complete jobs collected prior to the error are passed to the operating system for batch processing.

User Action: Reenter the command with the proper operands or enter another command. Check the command format to determine how many operands are allowed on the command that was in error.

If SUBMIT was specified, remember not to resubmit any job that was completely collected prior to the error.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK335 MSG SAVED FOR LOGON

Explanation: The message to be sent as a result of the SEND command just entered will be delivered when the specified recipient next logs on, since he is currently inactive.

System Action: The message is retained for delivery, and the user is prompted for another command.

User Action: Continue with your next command.

IHK336 INVALID DSNAME dsname [userid][key][linenum]

Explanation: The dsname specified on the last command entered is not valid. A CRJE data set name must contain 1-8 alphameric characters, the first of which must be alphabetic. The fully qualified name, which must not exceed 44 characters, must be specified for an OS data set.

If the last command entered was SUBMIT, the dsname of the data set in error will be displayed. If specified with the dsname when entered, the userid and/or key will also be displayed. If a DSLIST was being submitted for job processing, the linenum of the DSLIST data set in error is displayed.

System Action: The command is not processed. If jobs were being submitted into the job stream, any complete jobs collected prior to the error are passed to the operating system for batch processing. (A job is considered complete when another JOB card is encountered.) The user is prompted for another command.

User Action: Reenter the command with the proper dsname, or enter a new command. Edit the DSLIST data set if necessary (via the LIST and CHANGE or Implicit subcommands). When resubmitting the job stream, remember not to resubmit any job that was completely collected prior to the error.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK337 MSG NOT SAVED

Explanation: The message to be sent as a result of a SEND command cannot be delivered because of an I/O error or because the system's message-holding capacity has already been reached.

System Action: The message to be sent is ignored, and the user is prompted for another command.

User Action: Continue with your session by entering command. Allow a few minutes for some messages to be delivered before reentering the SEND command. If the error recurs, notify the central operator.

IHK338 DATA SET NOT FOUND [dsname][userid][key][linenum]

Explanation: A data set specified on the last command entered, or one specified in a DSLIST used in the last command, could not be located. If the last command entered was SUBMIT, the dsname will be specified. The userid and/or key may be displayed if they were assigned to the data set when created. If a DSLIST was being submitted, the DSLIST data set is specified. The linenum of the DSLIST data set in error is also displayed.

The dsname may have been specified incorrectly, an OS data set may not have been cataloged, or the data set may have been deleted.

System Action: If the error is detected during the job submission, any complete jobs collected prior to the error are passed to the operating system for batch processing, and no more processing is done for the SUBMIT command. If the error is on any other command, the command is not processed. The user is then prompted for another command.

User Action: Correct the dsname and reenter the command, or enter a new command. If the error occurred in a job submission, submit only the portion of the job stream not already processed.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK339 DATA SET NOT PDS OR SEQUENTIAL

Explanation: The OS data set specified on the last command entered is neither a partitioned data set nor a sequential data set. A data set must have partitioned or physical sequential organization in order to be accessible through CRJE.

System Action: The last command entered is not processed, and the user is prompted for another subcommand.

User Action: Enter another command. The only way you can gain access to the data set in question is to submit a job that will handle it.

IHK340 NO LINES IN RANGE

Explanation: There are no lines with line numbers in the range specified in the subcommand just entered.

System Action: No processing is performed for the subcommand just entered, and the user is prompted for another subcommand.

User Action: Enter your next subcommand.

IHK341 TEXT NOT FOUND

Explanation: The text operand of the CHANGE subcommand just entered could not be found in the line or lines in which the text was to be replaced.

System Action: No processing is done for the CHANGE subcommand, and the user is prompted for another subcommand.

User Action: Determine the text in the lines and reenter the CHANGE subcommand correctly, or enter a new subcommand.

IHK342 OUT OF DISK SPACE ddname

Explanation: In building a job stream, the SUBMIT command entered referred to an OS data set for which there is insufficient disk space either to allocate the data set or to contain the entire job stream. The ddname of the data set in error is displayed.

System Action: Any complete jobs collected before the error are passed to the operating system for batch processing. The user is prompted for another command.

User Action: Reenter the command remembering that you should submit only that portion of the job stream that has not been processed. If the error occurs frequently, notify the central operator.

IHK343 INVALID INCREMENT - 0

Explanation: The increment specified on the INPUT subcommand just entered was zero, which is an invalid increment.

System Action: The INPUT subcommand is not processed, and the user is prompted for another subcommand.

User Action: Reenter INPUT with a positive increment, or enter a new subcommand.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and then notify the central operator.

IHK344 JOB NOT FOUND jobname

Explanation: There is no record of a job with the indicated jobname in the CRJE system.

System Action: No processing is performed for the last command, on which the jobname was specified. The user is prompted for another command.

User Action: Check that the jobname was correctly specified on the command, and if not, reenter the command with the proper name. If you are sure the jobname was correct, the central operator may be able to explain why the job was not found. You may continue with your session by entering another command. Notify the central operator if the problem recurs.

IHK345 ORIGINAL NO. OF BLOCKS IN CLIST EXCEEDED

Explanation: A CLIST data set has been updated and saved while it was being executed, and the new version contains more blocks of 10 records each than the original version.

System Action: Execution of the CLIST is terminated. None of the commands in the additional block(s) are processed.

User Action: You must enter your next command from the terminal. The new CLIST can be executed via another EXEC command.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK346 LINE EXISTS, CANNOT INSERT line number

Explanation: The indicated line, which was specified on the INPUT subcommand just entered, cannot be inserted in the active data set because a line already exists with that line number.

System Action: The INPUT subcommand is not processed, and the user remains in edit mode. He is prompted for another subcommand.

User Action: Enter another subcommand. If you wish to replace the existing line, reenter INPUT without the I operand. If you wish to insert lines following the existing line, you must specify the line number for the first line to be inserted.

IHK347 UNABLE TO OPEN dsname

Explanation: An OS data set with the indicated dsname was specified on the last command entered, but the data set cannot be opened for processing.

System Action: The last command entered is rejected, and the user is prompted for another command.

User Action: Enter another command. Notify the central operator about the problem. Notify him, too, if the problem recurs.

IHK348 MAX LINE NUMBER EXCEEDED

Explanation: The system cannot prompt the user for the next line because the line number would be greater than 99999999, the maximum line number allowed.

If the last command entered was RENUMBER, the maximum line number will be exceeded if RENUMBER proceeds.

System Action: The user is put in edit mode and is prompted for a subcommand. Any lines entered prior to the error have been put in the active data set.

If RENUMBER was the last command entered, the system waits for the user to enter his next subcommand after sending an underscore prompt.

User Action: Enter another subcommand. To make room for lines to be added at the end of the data set, either use the RENUMBER subcommand to renumber the lines using a smaller increment between line numbers, or shorten the existing data set and then renumber it.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK349 LINE NOT FOUND line number

Explanation: A line with the indicated line number does not exist in the active data set.

System Action: The subcommand in which the line number was specified is not processed. The user is prompted for another subcommand.

User Action: Enter another subcommand. Listing the active data set will indicate the correct line number for the desired line. If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK350 ENTER USERID

Explanation: Either an invalid userid or no userid was specified on the LOGON command just entered.

System Action: The system waits for the user to enter his userid. If the userid is valid, processing of the LOGON command continues. If the userid is not valid, the LOGON command is not accepted, and the communication line is disconnected if it is a switched line.

User Action: Enter only the 1-7 character userid assigned to you by the installation. Do not enter a blank or any other character before typing your userid.

IHK351 ENTER PASSWORD ■■■■■■■■

Explanation: Either no password or an invalid password was specified on the LOGON command just entered. The eight-character area with the three characters printed on top of each other is provided to prevent others from seeing your password.

System Action: The system waits for the user to enter his password. If the password is valid, processing of the LOGON command continues. If the password is invalid, the LOGON command is not accepted, and the communication line is disconnected if it is a switched line.

User Action: Enter the 1-8 character password assigned to you by the installation. Do not enter a blank or any other character before typing your password. If you do not wish to type over the printed characters, roll the carriage up manually rather than using any of the keys on the terminal.

IHK352 INVALID USERID OR USERID IN USE, LOGON TERMINATED

Explanation: The userid entered in response to a prompting message was either invalid or was already being used by an active user.

System Action: Processing of the LOGON command is terminated, and the user is not allowed access to the system. The communication line is disconnected if it is a switched line.

User Action: None. When you have a valid userid, which no one else is using, you may reinitiate the LOGON procedure.

IHK353 INVALID PASSWORD, LOGON TERMINATED

Explanation: The password entered in response to a prompting message was invalid. Either the password is not recognized by the system, or it is not the password associated with the userid already entered.

System Action: Processing of the LOGON command is terminated and the user is not allowed access to the system. The communication line is disconnected if it is a switched line.

User Action: None. When you know the correct password, you may reinitiate the LOGON procedure.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and then notify the central operator.

IHK354 LOGOFF time SESSION TIME xxxx MIN.

Explanation: A LOGOFF command or another LOGON command has been entered to end a user's session. The indicated time is the time, according to the 24-hour system clock, that the session ended. xxxx is the total number of minutes the user was logged on.

System Action: The user's session is terminated. If the command entered was LOGOFF and the terminal is connected by a switched communication line, the line is disconnected.

User Action: None.

IHK355 USERID IN USE, ENTER USERID

Explanation: The userid specified on the LOGON command just entered is already being used by an active user.

System Action: The system waits for the user to enter another userid. If the user enters a valid userid, which is not currently in use, processing of the LOGON command continues. If the user does not enter an acceptable userid, the LOGON command is not accepted and the communication line is disconnected if it is a switched line.

User Action: If there is another userid you can use, enter only that 1-7 character userid. Do not enter a blank or any other character before typing the userid. If there is not other userid you can use, enter the end-of-line sequence for your terminal type so that the system can terminate the LOGON procedure. You may log on later, when the user currently using your userid has logged off.

IHK356 INVALID SOURCE MARGINS

Explanation: The source margins specified on an EDIT command for a PL/I data set are in error. One of several errors could have occurred: 1) the left margin may have been greater than the right margin; 2) only one margin may have been specified; 3) the right margin may have been greater than 80 (or 72 if line numbers are maintained in positions 73-80); or 4) zero may have been specified as one of the source margins.

System Action: No processing is performed for the EDIT command and the user is prompted for another command.

User Action: Reenter the EDIT command with the proper source margins, or enter a new command.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK357 ACTIVE AREA OUT OF SPACE

Explanation: There is not enough room in the active area to perform the operation requested with the last command entered, or there is not enough room for the last line entered in input mode.

System Action: The last command entered is not processed, or if a line was just entered in input mode, it is not added to the active data set. The user is prompted for another command.

User Action: Allow a few minutes for other users to finish any processing that may involve the active area, and reenter the command. If you have done a number of editing operations on your active data set, then saving the active data set, entering the END subcommand, and re-editing the data set you saved may free some space in the active area. If the problem recurs, notify the central operator.

IHK358 MESSAGE LOST

Explanation: One or more messages for the user from CRJE, the central operator, or another user have been lost because of a communication line error, an accumulation of more messages than the system can maintain, or an I/O error on disk.

System Action: None. The user is prompted for his next command.

User Action: Continue your session by entering another command.

IHK359 INSERT TERMINATED. NEXT LINE NUMBER IS line number

Explanation: The user cannot insert any more lines in input mode because the next line number generated would be equal to or greater than the line number of an existing line. The displayed line number is the line number of the next existing line.

System Action: The user is returned to edit mode and is prompted for a subcommand. Those lines already entered have inserted in the active data set.

User Action: Enter another subcommand.

IHK360 INVALID SCAN REQUEST, DATA SET NOT PL1 or FORTRAN

Explanation: A syntax scan was requested for a data set that does not have one of the following attributes: PL1, FORTE, FORTG or FORTH. System Action: If the scan was requested with the SCAN subcommand, the subcommand is not processed and the user is prompted for another subcommand. If the scan was requested with the SCAN operand of the EDIT command, then the EDIT command is processed as if the SCAN operand had not been specified.

User Action: Continue with your session. The LISTDS command can be used to determine what attribute is assigned to the data set (the default is DATA). If the attribute is incorrect, it can be changed by using the EDIT command.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK361 END OF DATA

Explanation: The end of the data set has been reached. This message may also indicate there was no SYSOUT data for the job whose output was just requested.

System Action: This completes processing of the last command entered. The user is prompted for another command.

User Action: Enter your next command.

IHK362 TP LINE ERROR

Explanation: An error that cannot be corrected occurred on the teleprocessing line during the transmission of data to or from the user's terminal. This is the reason the user's previous session was terminated.

System Action: The system tried to save the user's active data set if he had one when the error occurred. Another message indicates whether or not the data set was saved.

User Action: Proceed with your session. If the problem recurs, notify the central operator.

IHK363 USER INACTIVE, MSG NOT SAVED

Explanation: The message to be sent as a result of the SEND command just entered will not be delivered because the specified recipient is currently inactive.

System Action: The user is prompted for another command.

User Action: Enter another command. If you wish the message to be sent to the user when he next becomes active, reenter the SEND command with the LOGON operand.

IHK364 LINE line number NOT IN RANGE OF LINES SCANNED

Explanation: The subcommand just entered following a syntax error specified the displayed line number, which is not in the range of lines being scanned when the syntax error was detected.

System Action: The subcommand just entered is rejected, and the user is prompted for another subcommand to correct the error.

User Action: If the line number was incorrect, reenter the subcommand with the proper line number. If you do not wish to alter any more lines within the range of lines scanned, enter a null line in order to resume processing in input mode.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK365 INVALID JOBNAME jobname

Explanation: The displayed jobname, which was on a JOB card in the input stream specified by a SUBMIT command, contains more than eight characters. Positions 3-10 of the JOB card are displayed. This message is also sent if the jobname is missing from the JOB card.

System Action: Any jobs preceding the indicated job in the input stream are passed to the operating system for batch processing. The rest of the input stream specified by SUBMIT is ignored. The user is prompted for another command.

User Action: Enter another command. Before resubmitting the remainder of the job stream, correct the invalid jobname or insert the proper jobname by editing the data set containing invalid JOB card.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK366 DSNAME MUST BEGIN WITH USERID

Explanation: The DELETE command just entered requested the deletion of an OS data set for which the first qualifier of the dsname is not the user's userid.

System Action: The data set is not deleted, and the user is prompted for another command.

User Action: Continue with your session by entering another command. You cannot delete the data set by means of a CRJE command.

IHK367 INVALID LINE NUMBER RANGE

Explanation: The first line number specified for a range was greater than the second line number.

System Action: The subcommand containing the error is not processed, and the user is prompted for another subcommand.

User Action: Reenter the subcommand, specifying the lower line number first, or enter a new subcommand.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK368 TOO MANY CONTINUATION LINES. SCAN line number line number

Explanation: The installation-imposed maximum number of continuation lines has been collected and the indicated range of lines has been passed to the syntax checker for analysis.

System Action: The indicated lines are scanned for syntax errors. There may be error diagnostics since the statement collected was incomplete. After the indicated group of lines has been checked and any errors corrected, input mode is resumed if the lines were being entered in input mode. If the lines are part of a larger range of lines being checked as a result of the SCAN subcommand, after the lines are checked and any diagnostics printed, processing is continued by collecting the next group of lines in the statement, beginning with the first continuation line not scanned.

User Action: If the statement being entered requires more than the maximum number of continuation lines allowed by the system, you may ignore any errors resulting from scanning an incomplete statement. The problem will not exist when the statement is compiled.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK369 INVALID KEY {key
 {dsname [userid] [key] [linenum] }

Explanation: Either 1) an incorrect key or no key was specified on an EDIT command for a protected data set in another user's library, 2) the key specified on a subcommand was not 3 characters long, or 3) an incorrect key or no key was specified on a SUBMIT command for a protected data set. If either of the first two cases occurred, only the key is displayed with this message. If the third case occurred, the dsname of the data set in error, along with its userid and/or key (if applicable), is displayed. If a DSLIST has been submitted, the linenum (line

number) associated with the DSLIST data set in error is also displayed.

System Action: The command is rejected and the user is prompted for another command. If the error is detected during job submission, any complete jobs collected prior to the error are passed to the operating system for batch processing, and no more processing is done for the SUBMIT command.

User Action: Reenter the command with the correct key, or enter a new command. If a DSLIST was being submitted, edit the data set in error before reentering it with the other data sets. If the error was detected during a job submission, remember to submit only the portion of the job stream not already processed.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK370 LIBRARY INOPERATIVE

Explanation: There has been an I/O error on a user library referred to in the last command entered, which has caused it to be marked inoperative.

System Action: The command last entered is rejected and the user prompted for another command. There will be no attempt to read from or write to an inoperative user library until the CRJE system is closed down and restarted. A message is also sent to the central operator.

User Action: Notify the central operator. You may still enter commands that do not require access of the user library that is inoperative. Furthermore, if the problem recurs, notify the central operator.

IHK371 INVALID EXEC, DATA SET IS NOT CLIST

Explanation: The data set specified on the EXEC command just entered does not have the CLIST content attribute.

System Action: The EXEC command is not processed, and the user is prompted for another command.

User Action: If the data set does contain a command list, then use EDIT and SAVE TO ASSIGN IT THE CLIST attribute and reenter the EXEC command. If the data set does not contain a command list, EXEC is invalid and you should enter another command.

IHK372 EXTRA COMMAS IGNORED

Explanation: In the last command entered, one or more commas were used between the operation and operand fields, or two or more commas were used between operands.

System Action: The additional commas are ignored and the command is processed.

User Action: None. You should use only blanks between the operation and operand fields, and only one comma and any number of blanks as a delimiter between operands.

IHK373 LINE TOO LONG, REENTER LINE

Explanation: The last line entered, which may have been a command or a line entered in input mode, contained more than 134 characters counting backspace characters, blanks, and correction

characters.

System Action: The input is lost. The prompt is not reissued, and the system waits for the line to be reentered.

User Action: Reenter the input, breaking it into two lines or more if necessary.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

If this problem recurs three times consecutively (i.e., if you try to enter the line of input three times in succession and you receive this message twice) your session will be terminated. You will have to log on again, following the procedure for your particular terminal type.

IHK374 NO DISCONTINUED OUTPUT AVAILABLE

Explanation: A CONTINUE command has been entered, but no job output has been interrupted since the last OUPPUT command entered during this session.

System Action: The CONTINUE command is rejected, and the user is prompted for another command.

User Action: If the output of the desired job was interrupted during a previous session, or another OUTPUT command has been entered since the interruption, then use the OUTPUT command to obtain the remaining output of the job. The STATUS command may be used to determine if output for the job is still in the system.

IHK375 NO DD STATEMENT IN PROCEDURE OR LIBRARY UNAVAILABLE

```
[ { ddname }  
  { dsname } [userid][key][linenum] ]
```

Explanation:

OS Data Set - the ddname is displayed. The last command entered referred to an OS data set on a direct-access storage device for which there was no DD (Data Definition) statement included in the CRJE startup procedure. No data set residing on that device is accessible until a DD statement is added to the procedure and the system is restarted.

CRJE Data Set - If the last command entered was SUBMIT, the dsname of the data set in error is displayed. If applicable, the userid and/or key is displayed. If a DSLIST data set was being submitted, the line number associated with the DSLIST data set in error is also displayed. As with an OS data set, there was no DD statement included in the CRJE startup procedure for the CRJE data set on a direct-access storage device.

This message is also sent if the user does not have a library because either 1) he has been added to the system but no library was created for him, or 2) his library has been scratched. As a result of either action, the user cannot enter a SAVE, LISTDS, or LISTLIB command.

System Action: For the OS and CRJE data sets where there was no DD statement in the CRJE startup procedure, the last command entered is not processed unless it was SUBMIT, in which case any complete jobs collected prior to the error are passed to the

operating system for batch processing. The user is prompted for another command.

If the user's library was unavailable, the SAVE, LISTDS, or LISTLIB command is not processed.

In either case, the user is then prompted for another command.

User Action: If there was no DD statement in the procedure for the desired data set, inform the central operator that you would like access to the subject device. When resubmitting the job stream, remember to submit only the portion of the job stream that has not been processed. Edit the DSLIST data set in error if needed, also.

If your library was unavailable, process the active data set, if possible, or list it so that you reconstruct it later. Notify the central operator library is not available or have a library created for your use.

IHK376 SUBMIT CANCELLED BY INSTALLATION xxxxxxxx

Explanation: When checking the JCL statements in the input stream specified by the SUBMIT command just entered, an installation-provided routine rejected the job, which has a JCL card containing the displayed characters represented by "xxxxxxx." These x's represent characters found in positions 3-10 of the card that actually caused the submission to be cancelled; these characters may be the jobname, blanks, or other data.

System Action: Any jobs preceding the rejected job are passed to the operating system for batch processing. The remainder of the input stream is not processed, and the user is prompted for another command.

User Action: Check with the central operator to find out why the JCL card was rejected. Then correct the card and resubmit the portion of the input stream that was not processed, or enter a new command.

IHK378 NO JOBS SUBMITTED

Explanation: The input stream described by the SUBMIT command just entered did not include a JOB card.

System Action: No input is passed to the operating system for processing. The user is prompted for another command.

User Action: Enter another command. Include a JOB statement in the input stream before reentering the SUBMIT command.

IHK379 INVALID CORRECTION COMMAND command

Explanation: The indicated command, which was entered following a syntax error diagnostic, is not an Implicit subcommand or a CHANGE subcommand, which are the only two subcommands accepted for correcting syntax errors detected in input mode.

System Action: The command just entered is rejected, and the user is prompted for a subcommand that will correct the syntax error.

User Action: Use the Implicit or CHANGE subcommand to correct the syntax error. If no corrections are to be made, enter a null line to reenter input mode. To reenter the rejected command, you must be in edit mode (entered via a null line in input mode) or command mode (entered via the END subcommand in edit mode.)

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK380 EXEC TERMINATED

Explanation: A system error or an error in the command list has caused the execution of the command list to be terminated. If the error was in a command in the CLIST, a message has already been sent identifying the specific error.

System Action: No more commands in the CLIST are executed. The user is prompted for a command to be entered from his terminal.

User Action: Enter the next command. If you were in edit mode when the error occurred, you must enter an EDIT subcommand. You may correct an error in a command contained in the CLIST by editing the CLIST data set.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK381 TABS EXCEED ALLOWED LIMIT

Explanation: Either a logical tab greater than 80 was specified for input or a physical tab greater than 120 was specified for output on the TABSET command just entered.

System Action: The TABSET command is processed for tab settings specified up to the allowed limit. The user is prompted for another command.

User Action: Enter another command. You may wish to clear the existing tabs, since only 80 characters are allowed for input and 120 for output in line entered in input mode.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK382 TABS NOT IN ASCENDING ORDER

Explanation: The numbers for the tab settings specified in the TABSET command just entered were not entered in order from smallest to largest.

System Action: The TABSET command is not processed, and the user is prompted for another command. Any previously specified logical tabs are not cleared.

User Action: Reenter TABSET, specifying the numbers for the tab settings in ascending order, or if the existing tabs are suitable, enter a new command.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and then notify the central operator.

IHK383 ROUTING CODE NOT 1-16

Explanation: The routing code specified with the OPERATOR keyword on the SEND command just entered is greater than 16.

System Action: The SEND command is rejected, and no message is sent. The user is prompted for another command.

User Action: Enter another command, or reenter the SEND command with the correct routing code or no routing code. If no routing code or if zero is specified, the message will be sent to the operator chosen by the installation to receive all such messages.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK384 DATA SET RESIDES ON MULTIPLE VOLUMES

Explanation: The OS data set specified on the last command entered does not reside on a single volume of a direct-access storage device.

System Action: The command is rejected, and the user is prompted for another command.

User Action: Enter another command. The only way you have access to a data set on multiple volumes is through background processing.

IHK385 STATUS jobname(SCHED (n)
EXECUTING
NORMAL END
ABNORM END
NOT QUEUED
DISK ERROR)

Explanation: The user has requested status information about the job with the displayed jobname or about all jobs in the system that he submitted. The possibilities are as follows:

1. SCHED (n) - The job is the nth job on the input queue of jobs in the same class; that is, there are n-1 jobs to be initiated before the indicated job is initiated.
2. EXECUTING - The job is currently running.
3. NORMAL END - The job has completed execution, terminating normally.
4. ABNORM END - The job has completed execution but did not terminate normally.
5. NOT QUEUED - The operating system has not finished reading the input stream specified in the SUBMIT command.
6. DISK ERROR - An I/O error occurred when trying to retrieve the status information.

System Action: CRJE processing continues with a prompt for the user's next command.

User Action: Continue with your next command. Output is available if the job has completed. Notify the central operator when a disk error occurs or recurs.

IHK386 LEVEL OF SYNTAX CHECKER NOT AVAILABLE

Explanation: A FORTRAN syntax scan has been requested, but the syntax checker available is not for the language level specified in the content attribute for the data set.

System Action: If the scan was requested with a SCAN subcommand, the subcommand is rejected and the user prompted for another subcommand. If the automatic scan was invoked while the user was in input mode, the lines entered are not scanned and the user is prompted for another line of input.

User Action: Continue by entering your next subcommand or next line of input. If you can use the syntax checker that is available, use the EDIT command to change the attribute of the data set to the appropriate language level.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK387 SYNTAX CHECKER NOT IN SYSTEM

Explanation: A syntax scan has been requested for a data set in a language for which there is no syntax checker available.

System Action: If the scan was requested with a SCAN subcommand, the subcommand is rejected and the user prompted for another subcommand. If the automatic scan was invoked while the user was in input mode, the lines entered are not scanned and the user is prompted for another line of input.

User Action: Continue by entering your next subcommand or next line of input. The statements in the data set cannot be scanned.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK388 TIMEOUT

Explanation: The user waited too long between characters or before entering the first character, causing an intertext timeout. The last line entered has been lost.

System Action: The system waits for the line to be reentered. No prompt is issued.

User Action: Reenter the line.

IHK389 SESSION TERMINATED - ACTIVE AREA SAVED - DSNAME=ACTIVE

Explanation: The user's last session was terminated and his active data set was saved with the dsname ACTIVE in his user library. One of five conditions caused the active area to be saved: 1) CRJE abnormal closedown; 2) CRJE normal closedown; 3) line error; 4) line deactivated by central operator; or 5) startup after abnormal termination.

System Action: The system sends this message to the user if any of the five conditions occurs before the user logs off; otherwise, the message is sent when the user's session is reinitiated. After the message is sent (in the latter case), the user is prompted for a command.

User Action: Enter a command. You may resume processing where you left off by editing the data set ACTIVE.

IHK390 OUT OF SPACE ON LIBRARY DURING ACTIVE AREA RECOVERY

Explanation: The user's last session was terminated abnormally, but his active data set was not saved because there was insufficient space available in his user library. There was no attempt to condense the library.

System Action: The user is prompted for a command.

User Action: Enter a command. In order to resume processing where you left off, you must reconstruct the active data set.

IHK391 INVALID EDIT SUBCOMMAND command

Explanation: The indicated command, which was just entered, is not valid in edit mode.

System Action: The command rejected, and the user is prompted for a EDIT subcommand.

User Action: Enter an EDIT subcommand. To return to command mode, enter the END subcommand, which also deletes the active data set.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK392 INVALID SAVE, NO LINES IN DATA SET

Explanation: The user has attempted to save his active data set, but there are no lines in the data set.

System Action: No processing is performed for the SAVE subcommand, and the user is prompted for another subcommand.

User Action: Enter another subcommand.

IHK393 EDIT OF DATA SET REQ'D dsname

Explanation: The last command entered referred to the indicated CRJE data set for which no attributes or no line numbers have been assigned, for the data set was not created via CRJE and has not been edited.

System Action: No processing is performed for the last command entered. The user is prompted for another command.

User Action: Enter another command. Attributes and line numbers are assigned by specifying the data set in the EDIT command, although they are not permanently assigned until the edited data set is saved.

IHK394 DUPLICATE DSNAME, ENTER DSNAME

Explanation: The user's library already contains a data set with the dsname specified for the active data set being saved. The active data set is either a new data set, or the name specified on SAVE is not the same name specified on EDIT

System Action: The system waits for the user's response before continuing processing of the SAVE subcommand.

User Action: To replace the existing data set with the active data set, enter a null line. Enter only a 1-8 character dsname if you wish to change the name specified for the active data set.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK395 LIBRARY FULL, ENTER DSNAME FOR DELETION

Explanation: There is not enough room in the user's library to save his active data set. The user may specify a CRJE data set to be deleted from his library to make room for his data set.

System Action: The system waits for a response from the user. If the user specifies a dsname for deletion, the system continues to prompt for data sets to be deleted until the user indicates he is through. If the user requests the deletion of any data sets, they are deleted, the library is condensed, and SAVE is retried. If the user does not request that a data be deleted, processing of the SAVE subcommand is terminated and the active data set is not saved in the user's library.

User Action: To delete a data set, enter only the 1-8 character name of the data set to be deleted. If no data sets are to be deleted, enter a null line.

IHK396 ENTER DSNAME FOR DELETION

Explanation: The dsname just specified for deletion has been accepted, and the system is ready for the user to specify another data set for deletion.

System Action: The system waits for a response from the user. If the user specifies another dsname for deletion, prompting continues. If the user enters a null line, the user's library is condensed and the SAVE is retried.

User Action: If you are through deleting data sets, enter a null line. If you wish to delete another data set, enter only the 1-8 character dsname.

IHK397 DSNAME INVALID/MISSING, ENTER DSNAME

Explanation: The data set name entered in response to the preceding prompt was either an invalid dsname, or a data set with that name does not exist in the user's library.

System Action: The system waits for a response from the user. If the user enters a dsname, prompting continues. If the user enters a null line and other data sets have been deleted, his library is condensed and the SAVE retried. If he enters a null line and no data sets have been deleted, processing of the SAVE subcommand is terminated and the active data set is not saved in the user's library.

User Action: Enter only the correct 1-8 character dsname of the data set to be deleted, or enter a null line to indicate that you are through deleting data sets from your library.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

library or the last eight characters of the record if you have an 80-character record user library, must contain all numerics or all x's. This message will appear only once, even if each record in the data set contains invalid line numbers in the first, or last, eight positions (depending on the type of user library).

System Action: The incorrect line number is changed to all zeros, and the line is included in the active data set.

User Action: When the active data set is complete, you should list the data set to determine which lines were affected and renumber the lines before working with the data set.

IHK403 SESSION TERMINATED - ACTIVE AREA LOST

Explanation: The user's last session was terminated, and his active data set was lost because of one of the following five reasons: 1) there was an active area I/O error; 2) there was a library I/O error; 3) his library directory was full; 4) his library was full; or 5) there was a null data set.

System Action: The system sends this message to the user if any of the five conditions occurs before the user logs off; otherwise, the message is sent when the user's session is reinitiated. After the message is sent (in the latter case), the user is prompted for a command.

User Action: Enter a command. To resume processing where you left off, you must reconstruct the active data set.

IHK404 EXPIRATION DATE NOT REACHED - DATA SET NOT DELETED

Explanation: The OS data set specified in the DELETE command just entered could not be deleted because the expiration date assigned to the data set has not been reached.

System Action: The user is prompted for another command.

User Action: Enter another command. You cannot delete the data set via the DELETE command until the expiration date has been reached.

IHK405 LIBRARY EMPTY

Explanation: There are no CRJE data sets in the library of the user who entered the LISTLIB command.

System Action: The user is prompted for another command.

User Action: Enter another command.

IHK406 TP LINE ERROR, REENTER LINE

Explanation: An error on the communication line connecting the user's terminal to the central system has resulted in the loss of the last line entered.

System Action: The system waits for the line to be reentered. No prompt is issued.

User Action: Reenter the line. If the problem recurs frequently, notify the central operator about your problem.

IHK407 OPERATOR FAILED TO SPECIFY CORRECT PASSWORD

Explanation: The OS data set specified on the EDIT command just entered is password-protected, and the central operator did not supply the correct password, which is required to open the data set for processing.

System Action: Processing of the EDIT command is terminated, and the user is prompted for another command. He is still in command mode since no active data set was created.

User Action: Enter another command. Notify the operator of the correct password before trying to edit the protected data set again.

IHK408 INVALID EDIT REQUEST

Explanation: The user has attempted to edit a data set named ACTIVE in another user's library.

System Action: The EDIT command is not accepted, and the user is prompted for another command.

User Action: Enter another command. The only data set named ACTIVE to which you have access is the one in your own library.

If you are certain that you have entered the data correctly and the problem still recurs, retain the terminal listing and notify the central operator.

IHK409 DIRECTORY FULL, ENTER DSNAME FOR DELETION

Explanation: There is not enough room in the user's directory to save his active data set. The user may specify a CRJE data set to be deleted from his directory to make room for his data set.

System Action: The system waits for a response from the user. If the user specifies a dsname for deletion, the system continues to prompt for data sets to be deleted until the user indicates that he is through. If the user requests the deletion of any data sets, they are deleted, and SAVE is retried. If the user does not request that a data set be deleted, processing of the SAVE subcommand is terminated and the active data set is not saved in the user's directory.

User Action: To delete a data set, enter only the 1-8 character name of the data set to be deleted. If no data sets are to be deleted, enter a null line.

IHK410 DATA LOST IN TRUNCATION

Explanation: An attempt has been made to save records into an 80-character record user library wherein data, as opposed to blanks or the line sequence number, was found in positions 73-80. With an 80-character record user library, the input is limited to 72 characters.

System Action: The data in the last eight positions (73-80) is replaced by the line sequence number. The user is then prompted for his next subcommand, if possible. If CRJE has closed down normally or abnormally, the message is sent after CRJE has restarted and after message IHK389 has been sent to the user.

User Action: If the data in the last eight positions is needed to be saved, reenter the record on two lines, if necessary, restricting each record to no more than 72 characters. If the data in positions 73-80 need not be saved, enter your next subcommand.

IHK413 DIRECTORY FULL

Explanation: The user has issued an EDIT OLD command for a non-CRJE data set in his user library and insufficient space exists in his directory to expand the directory entry to CRJE format.

System Action: The data set is placed in the active area, and edit mode is entered.

User Action: If you wish to save the data set, after entering the SAVE subcommand, you will be prompted to delete data set names from your user library (IHK409 DIRECTORY FULL, ENTER DSNAME FOR DELETION). At this time, you may delete as many data sets as needed until there is ample space in your library to add this data set. If you do not want to edit or save the data set, enter the END subcommand to terminate edit mode.

Note: All of the keys of the 2741 keyboard referred to in this appendix are shown in Figure 3.

TURNING ON THE TERMINAL

1. Set the LCL/COM switch on the left side of the terminal stand to COM.
2. Press the ON portion of the ON/OFF key on the terminal keyboard.

END-OF-LINE SEQUENCE

Pressing the RETURN key generates the end-of-line sequence, which must be used to indicate the end of every line of either command input or data input.

ENTERING A LINE

1. Type the line as you would on an ordinary typewriter. The system automatically translates lowercase alphabetic to upper case unless the input is for a data set with the TEXT content attribute.
2. Enter the end-of-line sequence.

Note: If the first five characters of a line are nine's, a terminal test may be invoked and the line will be lost. To avoid this, backspace before entering the initial nine's.

CANCELLING A LINE

Before entering an end-of-line sequence, you can cancel the line being entered by depressing the BACK SPACE key once, immediately followed by the end-of-line sequence. The system will then expect a line to be entered without additional prompting.

CORRECTING A LINE

Before entering an end-of-line sequence, you can correct errors in the line being entered by backspacing to the first incorrect character and reentering the line from that point on. The characters and spaces entered replace the originally typed characters and spaces. To prevent overprinting, you may wish to manually move the paper up one or two lines.

INTERRUPTING OUTPUT

The procedure for interrupting output being received at a 2741 terminal depends upon whether the terminal is equipped with the special Interrupt feature.

WITH INTERRUPT FEATURE

1. Press the attention (ATTN) key on the terminal keyboard when you want to stop the output.
2. Enter the next command.

Note: If, after pressing the ATTN key, you wish to resume output rather than enter another command, enter one space followed by the end-of-line sequence, or enter a null line.

WITHOUT INTERRUPT FEATURE

After each group of lines is sent to the terminal without the Interrupt feature, the system waits for you to indicate the action to be taken. The possibilities are:

1. To receive the next group of lines, enter a null line.
2. To receive the remaining output ready to be delivered, enter one space followed by the end-of-line sequence.
3. To stop receiving the output, enter the next command.

ERROR CONDITIONS

- If the keyboard is locked for an extended period of time, the communication line connection may be broken. If depressing the ATTN key does not invoke a system response, there has been a line error and your session has been terminated.
 1. Reset the terminal by pressing the OFF portion of the ON/OFF key.
 2. Turn the terminal back on, dial the system if necessary, and reinitiate your session. If the error persists, have the communication line checked.
- If a line error occurs while entering input, you will receive the message, IHK406 TP LINE ERROR, REENTER LINE. You may then try twice to reenter your data.
- If an output error occurs, the system will try twice to retransmit the data. If the error is not corrected, you are automatically logged off. Your data set in the active area will be saved in your user library with a data set name of ACTIVE.
- If you enter a line of data that contains more than 134 characters, you will receive the message, IHK373 LINE TOO LONG, REENTER LINE. The line will be lost and no prompt will be reissued. You must reenter the data in either a shorter version or in two lines, if necessary.

ON-LINE TERMINAL TEST

The BTAM On-Line Terminal Test facility provides tests that can be used by the terminal operator as a start-up procedure or by the customer engineer for terminal checkout and diagnosis of terminal failure. The installation determines whether or not the On-Line Terminal Test facility is provided.

The facility may be used at logon time only, after the system has sent an underscore prompt, but before the LOGON command is entered. The system will not send another underscore prompt after testing but will wait for the LOGON command to be entered.

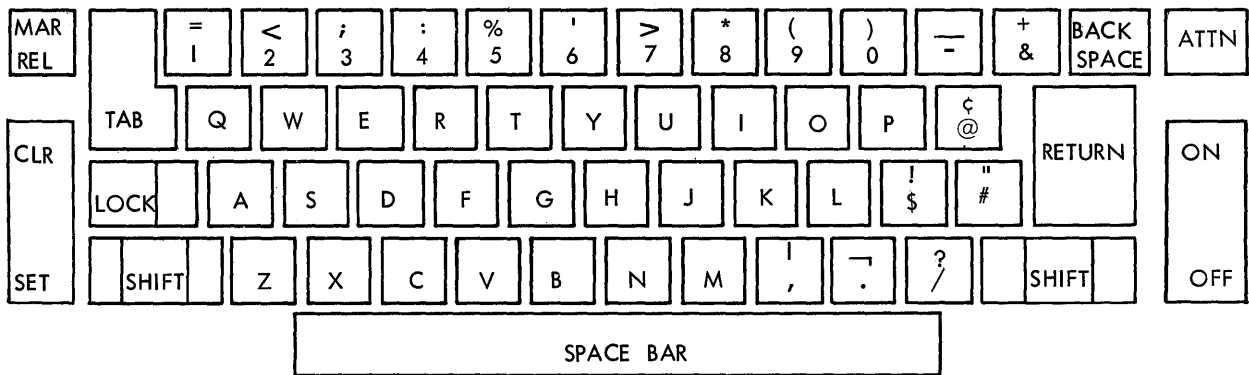


Figure 3. IBM 2741 Communications Terminal Keyboard

Note: All of the keys of the 2740 keyboard referred to in this appendix are shown in Figure 4.

TURNING ON THE TERMINAL

1. Press the COM portion of the LCL/COM key on the terminal keyboard.
2. Press the ON portion of the ON/OFF key on the terminal keyboard.

END-OF-LINE SEQUENCE

How the end-of-line sequence is generated depends upon whether the terminal is equipped with the Automatic EOB feature. This feature causes the end-of-block (EOB) line control character to be generated automatically when the RETURN key is depressed, provided the Auto Ck switch on the left side of the terminal stand is set to Mplx.

WITH AUTOMATIC EOB

Pressing the RETURN key generates the end-of-line sequence, which must be used to indicate the end of every line of either command input or data input.

WITHOUT AUTOMATIC EOB

The end-of-line sequence is generated by depressing the RETURN key and the EOB key. If the carriage need not be returned (e.g., entering a null line), the EOB key is sufficient.

Programmer's Note: A carrier return character immediately preceding EOB is deleted.

ENTERING A LINE

1. If the Transmit (T) light is not on, press the BID key.
2. When the Transmit (T) light comes on, type the line as you would on an ordinary typewriter. The system automatically translates lowercase alphabets to upper case unless the input is for a data set with the TEXT content attribute.
3. Enter the end-of-line sequence. The transmit light goes off, and the Restart (RST) light goes on briefly. If the Transmit light comes back on, you can enter another line. If the system sends you any output, you must use the BID key the next time you enter a line.

Note 1: If a line error occurs while you are entering input, you will receive the message, IHK406 TP LINE ERROR, REENTER LINE. You may try twice to reenter your data.

Note 2: If the first five characters of a line are nine's, a terminal test may be invoked and the line will be lost. To avoid this, backspace before entering the initial nine's.

CANCELLING A LINE

Before entering an end-of-line sequence, you can cancel the line being entered by depressing the BACK SPACE key once, immediately followed by the end-of-line sequence. The system will then expect a line to be entered without additional prompting.

CORRECTING A LINE

Before entering an end-of-line sequence, you can correct errors in the line being entered by backspacing to the first incorrect character and reentering the line from that point on. The characters and spaces entered replace the originally typed characters and spaces. (To prevent overprinting, you may move the paper up one or two lines. The paper may be moved via the INDEX key or by hand.)

INTERRUPTING OUTPUT

After each group of lines is sent to the terminal, the system waits for you to indicate the action to be taken. The possibilities are:

1. To receive the next group of lines, enter a null line.
2. To receive the remaining output ready to be delivered, enter one space followed by the end-of-line sequence.
3. To stop receiving the output, enter the next command.

ERROR CONDITIONS

- If a line error occurs while you are entering input, you will receive the message IHK406 TP LINE ERROR REENTER LINE. You may try twice to reenter your data.
- If you use the LOCK key for capitalization when entering data, be sure to unlock the keyboard before you enter the end-of-line sequence. Otherwise, a line error occurs. (The same message -- IHK406 -- and response apply here.)
- If an output error occurs, the system will try twice to retransmit the data. If the error is not corrected, you are automatically logged off. Your data set in the active area will be saved in your user library with a dsname of ACTIVE. If you are logged off, you would follow this procedure:
 1. Reset the terminal by depressing the OFF portion of the ON/OFF key.

2. Turn the terminal back on, dial the system if necessary, and reinitiate your session. If the error persists, have the communication line checked.
- If you enter a line of data that contains more than 134 characters, you will receive the message, IHK373 LINE TOO LONG, REENTER LINE. The line will be lost and no prompt will be reissued. You must reenter the data in either a shorter version or in two lines, if necessary.

ON-LINE TERMINAL TEST

The BTAM On-Line Terminal Test facility provides tests that can be used by the terminal operator as a start-up procedure or by the customer engineer for terminal checkout and diagnosis of terminal failure. The installation determines whether or not the On-Line Terminal Test facility is provided.

The facility may be used at logon time only, after the system has sent an underscore prompt but before the LOGON command is entered. The system will not send another underscore prompt after testing but will wait for the LOGON command to be entered.

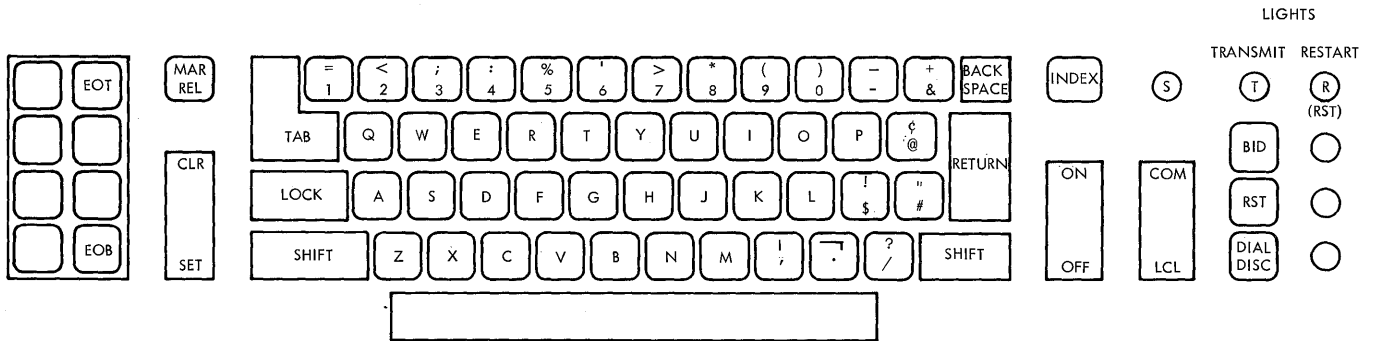


Figure 4. IBM Communications Terminal Model 1 Keyboard with Record Checking Feature

TURNING ON THE TERMINAL

1. Set the panel switches on the 1052 Printer-Keyboard as follows:

<u>Switch</u>	<u>Setting</u>
SYSTEM	ATTEND
PRINTER 1	SEND REC
KEYBOARD	SEND
SYSTEM	not on DIAL DSC
TEST	OFF
SINGLE CY	OFF
RDR STOP	OFF

Any other printer, reader, or punch switches, the STOP CODE switch, and the AUTO FILL switch should be set to OFF. The PUNCH and SYSTEM (PROGRAM/DUP) switches can be set either way. (See Figure 5.)

2. Set the power switch located on the side of the 1051 Control Unit to POWER ON. If the DATA CHECK light is on, turn it off by depressing the DATA CHECK key.

Note: If the power switch of the terminal on a leased line is in the OFF position when the START CRJE command is issued, use the following procedure:

1. Turn the power ON.
2. Notify the operator to activate the line (CRJE MODIFY command).
3. Depress the REQUEST key if your terminal has one.
4. Begin the LOGON sequence when the PROCEED light comes on.

END-OF-LINE SEQUENCE

How the end-of-line sequence is generated depends upon whether the terminal is equipped with the Automatic EOB feature. This feature causes the end-of-block (EOB) line control character to be generated automatically when the RETURN key is depressed, provided the EOB switch is set to AUTO.

WITH AUTOMATIC EOB

Pressing the RETURN key generates the end-of-line sequence, which must be used to indicate the end of every line of either command input or data input.

SYSTEM	MASTER	PRINTER 1	PRINTER 2	KEYBOARD	RDR 1	RDR 2	PUNCH 1	PUNCH 2	STOP CODE	AUTO FILL	PUNCH	SYSTEM	EOB	SYSTEM	TEST	SINGLE CY	RDR STOP
Attend	On	Rec	Rec	Send	Send	Send	Rec	Rec	Sense	On	Normal	Program	Manual		On	Line	Line
		Send Rec.	Send Rec.	Off	Off	Off	Off	Off								Off	Off
Un-attend	Off	Home	Home	Home	Home	Home	Home	Home	Off	Off	Bksp	Dup	Auto	Dial Dsc.	Off	Home	Home

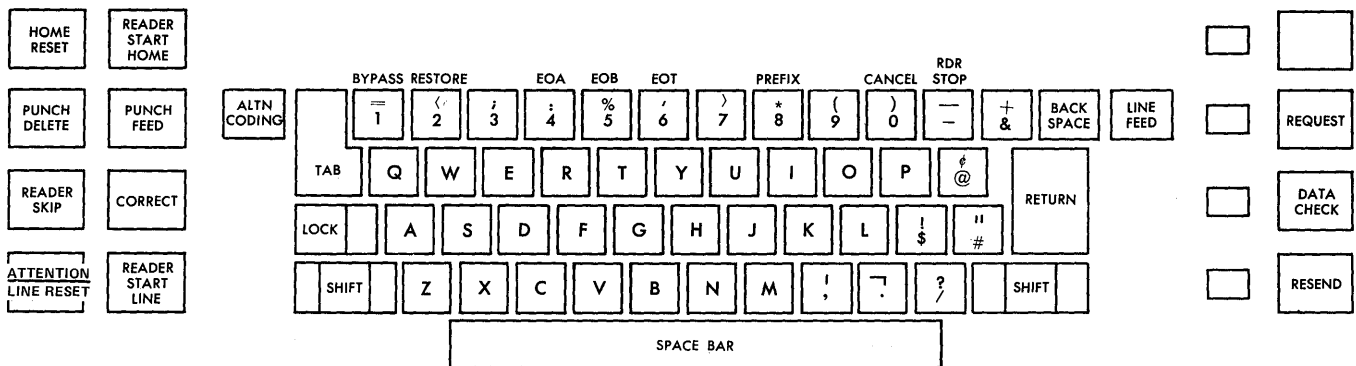


Figure 5. IBM 1052 Printer-Keyboard with Panel Switches (x = Proper Switch Setting)

WITHOUT AUTOMATIC EOB

The end-of-line sequence is generated by depressing the RETURN key and the EOB key (ALTN CODING key and 5 key). If the carriage need not be returned (e.g., when entering a null line), the EOB key is sufficient.

Programmer's Note: A carrier return character immediately preceding EOB is deleted.

ENTERING A LINE

The procedure for entering a line depends upon whether the 1052 Printer-Keyboard is equipped with the Keyboard Request feature, which provides an additional key, the REQUEST key.

WITH A REQUEST KEY

1. If the green PROCEED light is not on, press the REQUEST key.

2. When the PROCEED light comes on, type the line as you would on an ordinary typewriter. The system automatically translates lowercase alphabets to upper case unless the input is for a data set with the TEXT content attribute.
3. Enter the end-of-line sequence. The PROCEED light goes off, and the RESEND light goes on briefly. If the PROCEED light comes back on, you can enter another line. If the system sends you any output, you must use the REQUEST key the next time you enter a line.

Note: If the first five characters of a line are nines, a terminal test may be invoked and the line will be lost. To avoid this, backspace before entering the initial nines.

WITHOUT A REQUEST KEY

1. When the PROCEED light comes on, type the input line as you would on an ordinary typewriter. The system automatically translates lowercase alphabets to upper case unless the input is for a data set with the TEXT content attribute.
2. Enter the end-of-line sequence. The PROCEED light goes off, and the RESEND light goes on briefly. When the PROCEED light comes back on, enter your next line.

Note: Same as note described above.

CANCELLING A LINE

Before entering an end-of-line sequence, you can cancel the line being entered by depressing the BACK SPACE key once, immediately followed by the end-of-line sequence. The system will then expect a line to be entered without additional prompting.

CORRECTING A LINE

Before entering an end-of-line sequence, you can correct errors in the line being entered by backspacing to the first incorrect character and reentering the line from that point on. The characters and spaces entered replace the originally typed characters and spaces. (To prevent overprinting, you may move the paper up one or two lines. This may be done with the LINE FEED key or by hand.)

INTERRUPTING OUTPUT

The procedure for interrupting output being received at a 1050 terminal depends upon whether the terminal is equipped with the Keyboard Request feature.

WITH A REQUEST KEY

No action is required between groups of lines to continue receiving the output.

To interrupt the output:

1. Depress the REQUEST key at some time before the break in output.
2. When the PROCEED light comes on, type in the next command. This light will not come on until the break in output.

Note: If, after having pressed the REQUEST key, you wish to resume output rather than enter another command, enter one space followed by the end-of-line sequence when the PROCEED light comes on. After this action, there will be no more breaks or opportunities to interrupt the output.

WITHOUT A REQUEST KEY

While the KEYBOARD switch is set to SEND, the PROCEED light will come on after each group of "n" lines of output has been sent. You must then indicate your desired action by doing one of the following:

1. To receive the next group of lines, enter a null line.
2. To receive the remaining output ready to be delivered, enter one space followed by the end-of-line sequence.
3. To stop receiving the output, enter the next command.

Performing any one of these three actions may be bypassed by setting the KEYBOARD switch to OFF. This will cause your output to be sent with a slight pause after each group of "n" lines. Remember to reset this switch to SEND once your desired output has been transmitted.

ERROR CONDITIONS

Error conditions are indicated by lights on the 1052 Printer-Keyboard.

RECEIVE ALARM: This light is turned on if there is an incorrect switch setting, if paper is required, or if paper is not properly inserted.

DATA CHECK and RESEND: On together if input is being sent and a line error occurs. You may try twice to reenter the line. First, depress the DATA CHECK and RESEND keys to turn off the respective lights and reenter the line. If unsuccessful, your session is automatically terminated. You will have to LOGON again. (If on a leased line, this may require notifying the operator to activate the line.) If the error persists, have the communication line checked.

Note: You should never depress the RESEND key except when powering up the terminal or when it is used in conjunction with the DATA CHECK key when their respective lights are both on. (Usually when the 1050 is powered up, the RESEND light is on).

Only DATA CHECK is on if a line error occurs while receiving output. The system will try twice to retransmit the line. If the error is not corrected, your session is automatically terminated. You must depress

the DATA CHECK key along with the LINE RESET key to reset your communication line and logon again. If the error persists, have the communication line checked.

Note: If the 1050 on a leased line is turned off, the line must be reactivated by the central operator before logging on again.

TIME-OUT SUPPRESSION FEATURE

1050 WITH TIME-OUT SUPPRESSION

- With a leased line, you are allowed 1-1/2 minutes to enter a null line when preparing to logon. Otherwise, the line is deactivated. The operator must be notified to reactivate the line and you must try again.
- If you delay entering characters after the initial character has been entered, intertext time-outs are inhibited. The system will recognize a line of text only after the end-of-line sequence has been entered.

1050 WITHOUT TIME-OUT SUPPRESSION

- You are allowed 15 seconds to enter a null line when preparing to logon. Otherwise, the line is deactivated.
- If you wait more than 15 seconds before entering the first character of your line of data, the PROCEED light will go out. To correct this, reenter the line. (See "Entering a Line" for determining the proper procedure to follow.)
- If you delay more than 15 seconds between characters, you will get the message, IHK388 TIMEOUT, which means that the delay has caused an intertext time-out. The last line of input will be lost. You can correct this by reentering the line. The system will not reissue a prompt but will wait for the line to be reentered.

Note: If you enter a line of data that contains more than 134 characters, you will receive the message IHK373 LINE TOO LONG, REENTER LINE. The line will be lost and no prompt will be reissued. You must reenter the data in either a shorter version or in two lines, if necessary.

ON-LINE TERMINAL TEST

The BTAM On-Line Terminal Test facility provides tests that can be used by the terminal operator as a start-up procedure or by the customer engineer for terminal checkout and diagnosis of terminal failure. The installation determines whether or not the On-Line Test facility is provided.

The facility may be used at logon time only, after the system has sent an underscore prompt but before the LOGON command is entered. The system will not send another underscore prompt after testing but will wait for the LOGON command to be entered.

MESSAGE FORMAT

The format of a message produced by the PL/I Syntax Checker is:

IKMnnn llllllll text describing error

nnn is the identification number for the message. The message code, IKMnnn, will not appear unless the user specifically requests it when he logs on.

llllllll is the data set line number of the line in which the error was detected. Leading zeros and blanks are suppressed.

Text may contain the symbols xxx and yyy, which represent insertions from source input. The length of these insertions varies so that the maximum length of a message is 72 characters. An insertion is one syntactic unit with a maximum length of 15 characters.

USER ACTION

If an error is detected in input mode using the automatic scan facility, the user must enter a CHANGE or Implicit subcommand to correct the statement in error, or he must enter a null line to indicate that he does not wish to make a correction. Any corrections made must be in a line of the statement just scanned. After corrections are made, a null line returns the user to input mode and causes the statement to be scanned again.

If an error is detected in edit mode as a result of the SCAN subcommand, no specific user action is required. The user may correct errors at any time, using any of the EDIT subcommands.

ERRORS DETECTED

THOROUGH CHECKING

The PL/I Syntax Checker will detect almost all errors in a PL/I statement that are dependent on information present in a single statement. Examples of the errors found by thorough checking are:

- Unmatched string quotes, parentheses, and comment brackets,
- Unrecognizable statement (i.e., the main part of the statement, excluding labels and prefixes, does not start with an identifier or a semicolon),
- Errors in prefixes and labels,
- Errors in constants and pictures,
- Errors in format lists and data lists,
- Invalid use of options and attributes,

- Errors in expressions,
- Missing or invalid use of comma, colon, parenthesis, equal sign, or semicolon.

Certain errors are not detected although they can be detected in a single statement. The reason is that either these errors are special cases of errors normally dependent on information from more than one statement or that they are infrequent errors which are complicated to detect. Examples of these are:

- Multiple declarations of one item within the same DECLARE statement,
- Inconsistent use of one item within the same statement, e.g., LABEL: A=LABEL + 1; A(1) = A + A(1,2,3);,
- Errors in sterling constants,
- An INITIAL attribute list specifies too few or too many initial values for an item,
- More than 64 parameters in a parameter list.

Note: The PL/I Syntax Checker ignores compile-time statements.

RESTRICTED CHECKING

When the level of checking is restricted, the PL/I Syntax Checker will detect a subset of the through checking errors (see preceding section). The errors found by restricted checking are:

- Unmatched string quotes, parentheses, and comment brackets,
- Unrecognizable statement or missing semicolon,
- Errors in prefixes, labels and constants,
- Invalid input characters,
- All errors in assignment, BEGIN, STOP, END, and EXIT statements, and in the IF clause part of an IF statement.

ERROR MESSAGES

IKM001 11111111 UNMATCHED STRING QUOTES IN THIS STATEMENT

Explanation: The quotation mark at the end of a character or bit string is missing. No more checking of the statement is performed.

IKM002 11111111 UNMATCHES COMMENT BRACKETS IN THIS STATEMENT

Explanation: The character pair /* marking the end of a comment is missing. No more checking of the statement is performed.

IKM003 11111111 STATEMENT NOT RECOGNIZED

Explanation: A statement (including prefixes) does not start with an identifier, semicolon, decimal integer, or left parenthesis; or, a statement (excluding prefixes) starts with a non-keyword identifier but does not contain an equal sign or does not have an equal number of left and right parentheses on the left of the equal sign.

IKM004 11111111 LABEL MISSING FROM xxx STATEMENT

Explanation: The label that should prefix an ENTRY, PROCEDURE, or FORMAT statement is missing.

IKM005 11111111 PREFIX NOT PERMITTED BEFORE xxx

Explanation: A condition prefix to an ELSE clause, or an ENTRY or DECLARE statement, is not permitted.

IKM006 11111111 CHECK/NOCHECK PREFIX NOT PERMITTED BEFORE THIS STMT

Explanation: The prefixes CHECK and NOCHECK may precede only a PROC or BEGIN statement.

IKM007 11111111 PREFIX OPTION FOLLOWS LABEL

Explanation: A condition prefix follows a label prefix. This is not permitted under the rules of PL/I. A condition prefix must always precede any label prefix.

IKM008 11111111 ILLEGAL STATEMENT FOLLOWS xxx

Explanation: xxx = ON: An on-unit consisting of an IF, ON, DO, RETURN, PROCEDURE, DECLARE, END, or FORMAT statement, or an ELSE clause, is not permitted.

xxx = IF or ELSE: A unit-1 or unit-2 branch of an IF statement consisting of an ON, DO, PROCEDURE, ENTRY, DECLARE, END, or FORMAT statement, or an ELSE clause, is not permitted. The illegal statement is not checked further.

IKM009 11111111 INVALID LABEL BEGNG xxx

Explanation: A label is not an (conditionally subscripted) identifier.

IKM010 11111111 INVALID PREFIX OPTION xxx

Explanation: A condition prefix contains an invalid condition name. The checking continues after the next colon.

IKM011 11111111 LABEL NOT PERMITTED BEFORE xxx

Explanation:

xxx = ON: the on-unit in an ON statement may not have a label prefix.

xxx = ELSE: a label to an ELSE clause must not precede the word ELSE.

IKM012 11111111 INCOMPLETE IF STATEMENT, THEN NOT FOUND WHEN EXPECTED

Explanation: In an IF statement, the keyword THEN, or the entire THEN clause, is missing. The checking of the statement is terminated.

IKM013 11111111 ERROR IN xxx STATEMENT BEGNG yyy

Explanation:

xxx = ALLOCATE, FREE, DECLARE, OPEN, CALL, DO, GET, or PUT statements. An invalid symbol may be contained in one of these statements.

xxx = BEGIN. The BEGIN statement is incorrectly written (may have the option ORDER or REORDER in PL/I Version 5).

xxx = THIS. Error in an unclassified statement. Checking of the statement is terminated.

IKM014 11111111 INVALID CHAR xxx PRECEDING yyy

Explanation: The character is not a PL/I character. Checking of this statement is terminated.

IKM015 11111111 INVALID CHAR IN BIT STRING BEGNG xxx

Explanation: A bit string contains a character other than 0 or 1.

IKM016 11111111 xxx MISSING AFTER yyy

Explanation: A delimiter (comma, colon, or right parenthesis) or an identifier is missing.

IKM017 11111111 MISSING EQUAL SIGN IN DO SPECIFICATION

Explanation: A Type-3 DO statement or repetitive specification must have the following general form: DO variable = expression [...]; . This error terminates the checking of the DO specification.

IKM018 11111111 SURPLUS COMMA AFTER xxx

Explanation: A comma should not separate the options in a GET, PUT, or DECLARE (ENVIRONMENT) statement.

IKM019 11111111 A LETTER IMMEDIATELY FOLLOWS CONSTANT BEGNG xxx

Explanation: A constant may only be followed by one of the following: any of several special characters, e.g., a blank or semicolon; an arithmetic operator; a comparison operator; or a bit-string operator. This could not be internally translated so no further checking of the statement is performed.

IKM020 11111111 INVALID PREFIX OPERATOR xxx PRECEDING yyy

Explanation: An expression begins with an operator other than \uparrow , +, - or (or an operand other than an identifier, a string, or a constant.

IKM021 11111111 CONSTANT BEGNG xxx IS TOO LONG

Explanation: A binary fixed-point or integer constant has more than 31 digits, or a decimal fixed-point or integer constant has more than 15 digits.

A binary floating-point constant has more than 53 digits in the mantissa part, or a decimal floating-point constant has more than 16 digits in the mantissa part.

IKM022 11111111 CONSTANT BEGNG xxx EXCEEDS MAXIMUM VALUE

Explanation: A floating-point constant exceeds the value 7.205,759,403,792,793E75, which is the highest value allowed.

IKM023 11111111 SOLITARY DECIMAL POINT FOUND IN OPERAND POSITION

Explanation: A period appears invalidly in an expression in an operand position.

IKM024 11111111 EXPONENT MISSING IN CONSTANT BEGNG xxx

Explanation: The letter E in a floating-point constant is followed by some character other than a digit, a plus or minus sign, or a decimal point.

IKM025 11111111 EXPONENT TOO LONG IN CONSTANT BEGNG xxx

Explanation: A binary floating-point constant has more than 3 digits in the exponent part, or a decimal floating-point constant has more than 2 digits in the exponent part.

IKM026 11111111 DECIMAL POINT IN EXPONENT OF CONSTANT BEGNG xxx

Explanation: The exponent in a floating-point constant contains a decimal point.

IKM027 11111111 CONSTANT BEGNG xxx HAS TOO MANY DECIMAL POINTS

Explanation: A fixed-point constant or the mantissa part of a floating-point constant contains more than one radix point. Sterling constants are not checked.

IKM028 11111111 CONFLICTING xxx OPTION

Explanation: The option named conflicts with a previously specified option.

IKM029 11111111 xxx OPTION REQUIRED

Explanation: The option named must be specified.

IKM030 11111111 INVALID OPTION BEGNG xxx

Explanation: The option named is not a valid option keyword.

- IKM031 11111111 EXPRESSION MISSING AFTER xxx
Explanation: An expression to the right of an equal sign or IF is missing. Checking of the statement is terminated.
- IKM032 11111111 INVALID ARGUMENT SPECIFIED FOR xxx
Explanation: The attribute or option named has an invalid argument or argument list.
- IKM033 11111111 ARGUMENT MISSING AFTER xxx
Explanation: The argument that must follow the attribute or option named is missing, or the argument list is empty.
- IKM034 11111111 MULTIPLE xxx yyy SPECIFIED
Explanation: The attribute or option named has been previously specified in this statement.
- IKM035 11111111 NO FILE SPECIFIED IN OPEN/CLOSE STATEMENT
Explanation: The FILE option in an OPEN or CLOSE statement is missing. No more checking of the statement is performed.
- IKM036 11111111 ILLEGAL USE OR INCORRECT FORMAT OF REFER OPTION
Explanation: The REFER option is used in an illegal context or is specified in an invalid format.
- IKM037 11111111 FILE ORGANIZATION MISSING FROM ENVIRONMENT OPTION
Explanation: The file organization is not specified in the ENVIRONMENT option.
- IKM038 11111111 CR OR DB IN FLOATING FIELD OF PICTURE BEGNG xxx
Explanation: The CR or DB symbols must not be specified for the exponent of a floating field in a PICTURE specification.
- IKM039 11111111 CONFLICTING xxx ATTRIBUTE
Explanation: The attribute named conflicts with a previously specified attribute. For DECLARE and ALLOCATE statements, checking goes on for the next level-one declare variable (i.e., if the conflicting attribute is inside a structure, no more checking of this structure is performed).
- IKM040 11111111 xxx ATTRIBUTE NOT PERMITTED IN THIS CONTEXT
Explanation: An iSUB dummy variable appears outside the context of a DEFINED attribute of a DECLARE statement; or the LIKE attribute is specified in an invalid context.

IKM041 11111111 ERROR IN FORMAT LIST BEGNG xxx

Explanation: Either 1) a format item does not start with an identifier or a decimal integer constant; 2) the identifier is not a valid format item keyword, or 3) the FORMAT item has an invalid format. Checking continues after the format list.

IKM042 11111111 xxx MUST HAVE ENTRY ATTRIBUTE

Explanation: The identifier named must be declared with an ENTRY attribute.

IKM043 11111111 INVALID xxx ATTRIBUTE SPECIFIED FOR yyy

Explanation: The attribute named is not a valid attribute keyword.

IKM044 11111111 ILLEGAL BASE FOR DEFINED ITEM xxx

Explanation: The item named is defined on an invalid base.

IKM045 11111111 ERROR IN INITIAL ATTRIBUTE LIST BEGNG xxx

Explanation: The INITIAL attribute list does not begin with a constant, a string, or a repetition factor; or a + or - sign is not followed by a constant.

IKM046 11111111 INVALID CHAR xxx IN PICTURE BEGNG yyy

Explanation: The picture specification identified contains one or more invalid characters.

IKM047 11111111 ILLEGAL USE OF CHAR xxx IN PICTURE BEGNG yyy

Explanation: The character identified is valid but improperly used in a picture specification.

IKM048 11111111 EXPONENT FIELD MISSING IN PICTURE BEGNG xxx

Explanation: In the picture specification for a floating-point number, the exponent field is missing.

IKM049 11111111 EXPONENT FIELD TOO LARGE IN PICTURE BEGNG xxx

Explanation: In the picture specification for a floating-point number, the exponent field has too many decimal positions.

IKM050 11111111 MORE THAN ONE SIGN CHAR IN PICTURE BEGNG xxx

Explanation: The numeric picture specification identified contains more than one (not drifting) sign character.

IKM051 11111111 INVALID SCALING FACTOR IN PICTURE BEGNG xxx

Explanation: In the picture specification for a fixed-point number, the scaling factor is invalidly represented.

- IKM052 11111111 INVALID USE OF SCALING FACTOR IN PICTURE BEGNG xxx
Explanation: A scaling factor may occur only in the picture specification for a fixed-point number.
- IKM053 11111111 NO SCALING FACTOR PARENTHESES IN PICTURE BEGNG xxx
Explanation: The parentheses enclosing the integer of a scaling factor in the picture specification identified are missing.
- IKM054 11111111 INVALID REPETITION FACTOR IN PICTURE BEGNG xxx
Explanation: The repetition factor in the picture specification identified is not an integer. A repetition factor in a picture specification must be a non-zero integer enclosed in parentheses.
- IKM055 11111111 MULTIPLE USE OF CHARACTER xxx IN PICTURE BEGNG yyy
Explanation: The characters E, K, and V may occur only once in a numeric picture specification.
- IKM056 11111111 NO DIGIT POSITION BEFORE EXPONENT IN PICTURE xxx
Explanation: A picture specification for a floating-point number makes no allowance for the digits of a mantissa preceding the delimiter E.
- IKM057 11111111 PICTURE BEGNG xxx LONGER THAN 255 CHAR
Explanation: The picture specification identified exceeds the maximum length of 255 characters.
- IKM058 11111111 STERLING CHAR FOUND IN NON-STERLING PICTURE xxx
Explanation: One of the characters 8, 7, 6, P, G, H, or M is found but the character G did not start this picture specification.
- IKM059 11111111 ILLEGAL LEVEL NUMBER xxx
Explanation: A major structure name must be declared with the level number 1. Minor structures must be declared with level numbers greater than 1. Level numbers must be decimal integers.
- IKM060 11111111 PRECISION EXCEEDS xxx FOR PICTURE BEGNG yyy
Explanation: The precision implied by the picture specification for a fixed or floating-point number exceeds the maximum default precision.
- IKM061 11111111 ILLEGAL ASTERISK AS SUBSCRIPT IN DEFINING LIST
Explanation: An asterisk is not allowed as a subscript in a defining list.

- IKM062 11111111 ELEMENT xxx IS ILLEGALLY DEFINED WITH ISUB
Explanation: iSUB variables may not be used to define a scalar variable on an array base.
- IKM063 11111111 ILLEGAL ISUB VALUE
Explanation: The value of an iSUB dummy variable is outside the range 1-32.
- IKM064 11111111 POSITION ATTRIBUTE ILLEGAL FOR DEFINED ITEM xxx
Explanation: A POSITION attribute may not be specified for a data item defined by iSUB variables on a base identifier.
- IKM065 11111111 SUBSCRIPTED BASE ILLEGAL FOR DEFINED ITEM xxx
Explanation: The base identifier on which a data item is defined may not be a subscripted name.
- IKM066 11111111 FORMAT LIST MISSING
Explanation: The format list in a GET, PUT, or FORMAT statement is missing.
- IKM067 11111111 FORMAT LIST CONTAINS NO DATA FORMAT ITEM
Explanation: The format list in a GET or PUT statement must include a format item for the data item being transmitted.
- IKM068 11111111 FORMAT ITEM xxx PERMITTED WITH OUTPUT ONLY
Explanation: The format item named may not be used in the format list of a GET statement.
- IKM069 11111111 FORMAT ITEM xxx IS INVALID
Explanation: The format item named uses invalid characters or is incorrectly written. Checking continues after the end of the format list.
- IKM070 11111111 NESTING OF FORMAT LIST EXCEEDS 20
Explanation: A format list in this statement is nested to more than 20 levels.
- IKM071 11111111 INVALID DATA ITEM BEGNG xxx
Explanation: The data item is not a valid identifier or is incorrectly written.
- IKM072 11111111 NO DATA SPECIFICATION OR CONTROL OPTION FOUND
Explanation: The data list and format list for a GET or PUT statement are missing.

- IKM073 11111111 ARRAY BOUND FOR xxx IS TOO LARGE
- Explanation: The upper and/or lower bound of one or more dimensions of an array exceeds the maximum value.
- IKM074 11111111 *BOUNDS ARE MIXED WITH NON* BOUNDS
- Explanation: Variable array bounds (denoted by asterisks) may not be mixed with non-variable bounds in a DECLARE statement.
- IKM075 11111111 LOWER BOUND GREATER THAN UPPER BOUND FOR xxx
- Explanation: An array is declared with a lower bound greater than the upper bound or with a single upper bound equal to or less than zero.
- IKM076 11111111 EXTERNAL NAME BEGNG xxx LONGER THAN SEVEN CHAR
- Explanation: An external name may not exceed seven characters in length.
- IKM077 11111111 IDENTIFIER BEGNG xxx IS TOO LONG
- Explanation: An identifier has more than 31 characters.
- IKM078 11111111 UNMATCHED PARENTHESES, xxx MISSING
- Explanation: The left and right parentheses in the current statement are unbalanced; one or more left or right parentheses are missing. The statement is not checked further.
- IKM079 11111111 LEFT PARENTHESIS REQUIRED AFTER xxx
- Explanation: The option named must be followed by one or more arguments enclosed in parentheses.
- IKM080 11111111 OPERAND MISSING BEFORE xxx
- Explanation: An identifier or expression preceding the item named is missing. This error terminates checking of the expression in question.
- IKM081 11111111 INSUFFICIENT ARGUMENT SPECIFIED FOR xxx yyy
- Explanation: The number of arguments specified in the statement named is insufficient.
- IKM082 11111111 ON CONDITION INVALID OR MISSING
- Explanation: An ON-condition is either missing or invalidly specified. Checking of the ON statement is terminated.
- IKM083 11111111 INVALID SET/IN CLAUSE SPECIFIED FOR xxx
- Explanation: The clause following the SET or IN option is invalid.

- IKM084 11111111 VARIABLE IN LOCATE STATEMENT INVALID OR MISSING
- Explanation: The LOCATE keyword is not followed by an unsubscripted, unqualified identifier.
- IKM085 11111111 FACTORING NOT PERMITTED ON ALLOCATE STATEMENT
- Explanation: An ALLOCATE statement contains a list of two or more factored variables.
- IKM086 11111111 POINTER QUALIFIER AFTER POINTER QUALIFIER OR SUBSCRIPT
- Explanation: A pointer in a pointer qualifier may not be pointer qualified or subscripted.
- IKM087 11111111 OPERAND MISSING AFTER xxx
- Explanation: An identifier or expression following the item named is missing. Checking is terminated.
- IHKM088 11111111 NO DIGIT POSITION IN PICTURE BEGNG xxx
- Explanation: A picture specification must contain at least one digit position.
- IKM089 11111111 INVALID OPERATOR xxx
- Explanation: The operator is not valid for use in an expression.
- IKM090 11111111 IDENTIFIER MISSING AFTER xxx
- Explanation: An identifier is missing after the item named.
- IKM091 11111111 OPERATOR MISSING BEFORE xxx
- Explanation: An operator is missing after the item named. This error terminates the checking of the expression in question.
- IKM092 11111111 ILLEGAL LEFT PART OF ASSIGNMENT STATEMENT
- Explanation: An invalid operand appears to the left of the assignment symbol. Checking of the statement is terminated.
- IKM093 11111111 INVALID DO SPECIFICATION
- Explanation: The keyword TO or BY is multiply used, or the DO specification contains an illegal symbol.
- IKM094 11111111 xxx MISSING IN yyy STATEMENT
- Explanation: For example, the keyword TO or the label in a GO TO statement is missing.

IKM095 11111111 SEMICOLON NOT FOUND WHEN EXPECTED

Explanation: The semicolon marking the end of a logically complete statement is missing, or no semicolon is found for the last statement of the input buffers.

IKM096 11111111 INVALID EVENT NAME BEGNG xxx

Explanation: In a WAIT or input/output statement, the event name specified for the EVENT option is invalid.

IKM097 11111111 BREAK CHAR INVALID AT BEGNG OF IDENTIFIER xxx

Explanation: An identifier may not begin with a break character.

IKM098 11111111 INVALID FORM OF INITIALIZATION FOR xxx

Explanation: The initialization specified in this statement does not comply with the rules of PL/I syntax.

IKM099 11111111 MORE THAN 3 LEVELS OF ENTRY NESTING

Explanation: The ENTRY attribute may not apply to more than three logical levels.

IKM100 11111111 INVALID REPETITION FACTOR IN INITIAL LIST

Explanation: The repetition factor, a decimal integer constant, must be enclosed in parentheses and must precede the value or character to which it relates.

IKM101 11111111 MORE THAN 63 TRUE LEVEL NUMBERS SPECIFIED FOR xxx

Explanation: A structure may have a maximum of 63 levels, including the major structure number 1.

IKM102 11111111 MORE THAN 32 DIMENSIONS SPECIFIED FOR xxx

Explanation: An array may have a maximum of 32 dimensions.

MESSAGE FORMAT

The format of a message produced by the FORTRAN Syntax Checker is:

IPDnnn xxxxxxxx yyyyyy text describing error

nnn is the identification number for the message. The message code, IPDnnn, will not appear unless the user specifically requests it when he logs on (by the MSGID operand).

xxxxxxx is the data set line number of the line in which the error was detected. Leading zeros and blanks are suppressed.

yyyyyy indicates the character(s), up to six, in error. This field may be omitted if it is not applicable to the error.

USER ACTION

If an error is detected in input mode using the automatic scan facility, the user must enter a CHANGE or Implicit subcommand to correct the statement in error, or he must enter a null line to indicate that he does not wish to make a correction. Any corrections made must be in a line of the statement just scanned. After corrections are made, a null line returns the user to input mode and causes the statement to be scanned again.

If an error is detected in edit mode as a result of the SCAN subcommand, no specific user action is required. The user may correct errors at any time, using any of the EDIT subcommands.

ERROR MESSAGES

IPD000 SYSTEM OR SYNTAX CHECKER FAILURE

Explanation: While scanning the last statement, the syntax checker encountered a condition that should not occur. It may be a hardware, syntax checker, or operating system error.

IPD002 UNRECOGNIZABLE STMNT OR MISSPELLED KEYWD

Explanation: The statement was not a recognizable FORTRAN statement type, or a keyword of six or fewer characters was misspelled. An assignment statement with errors to the left of the equal sign is unrecognizable.

IPD004 UNSIGNED INTEGER EXPECTED

Explanation: An invalid form, such as an integer constant preceded by a plus or minus sign, was encountered in the statement where (1) an unsigned integer is the only valid form, or (2) an unsigned integer or unsigned variable is the only valid form.

IPD006 EXPRESSION EXPECTED

Explanation: An expression is either missing or invalid in the statement where a valid one is expected. For example, each of these statements would receive this message:

1. A=
2. G (I,J) = C (I,)
3. X= Y+2.0*(**2+6.28)

The source characters in error in the message would be: none for statement 1; ")" for statement 2; "**2+6." for statement 3.

IPD008 POSSIBLY TOO MANY SUBSCRIPTS PRECEDE

Explanation: A list of eight or more (four or more in FORTRAN E) names within parentheses has been found following a symbolic name on the left side of the equal sign in an assignment statement or statement function definition. If the statement is a statement function definition, the message should be ignored.

IPD010 TOO MANY SUBSCRIPTS

Explanation: A list of eight or more (four or more in FORTRAN E) expressions within parentheses has been found following a symbolic name in a place where a subscripted variable reference would be valid.

IPD012) EXPECTED

Explanation: Either a required parenthesis was missing or there was no right parenthesis to match a left parenthesis.

IPD014 ARITH IF REQUIRES STATEMENT NUMBER LIST

Explanation: There was a missing or invalid statement number in the list of statement numbers after an arithmetic IF statement. (FORTRAN G and H only)

IPD016 INVALID EXPRESSION IN IF STATEMENT

Explanation: The expression within parentheses after the "IF" keyword of a logical or arithmetic IF statement was not recognizable as either a valid logical or arithmetic expression. (FORTRAN G and H only)

IPD018 UNRECOGNIZABLE STMT AFTER LOGICAL IF

Explanation: The statement following the "IF (logical expression)" was not a recognizable FORTRAN statement type, or there was something other than blanks between the right parentheses at the end of the logical expression and the start of the statement. Some errors in arithmetic IF statements will cause this message to be issued. (FORTRAN G and H only)

IPD020 NON-ZERO INTEGER EXPECTED

Explanation: The indicated numeric constant is zero, not an integer constant, or both.

IPD022 ILLEGAL STATEMENT AFTER LOGICAL IF

Explanation: The statement following the "IF (logical expression)" is not one of the statement types permitted after a logical IF. The statement cannot be a DO, another logical IF, any non-executable statement, nor any of the DEBUG statements. (FORTRAN G and H only)

IPD026 DATA SET NUMBER EXPECTED

Explanation: An input/output statement had neither an unsigned nonzero integer nor a variable name in the correct position for the data set reference numbers.

IPD028 LENGTH SPECIFICATION INVALID

Explanation: In a type-statement, a length specification was missing or was incorrect for the type of statement specified for the variable(s) or function. (FORTRAN G and H only)

IPD030 (EXPECTED

Explanation: A required left parenthesis is missing. This message is not issued for statements where parentheses are optional (assignment statements, for example), but is issued where a pair of parentheses is mandatory (as in FUNCTION, EQUIVALENCE, WRITE, etc.).

IPD032 NAME EXPECTED

Explanation: A required name is missing or is preceded by characters that cannot begin a name. For example, all these statements will receive this message:

```
FUNCTION (A,B,C)
DIMENSION, X(20,30)
DEFINE FILE 8(10,20,U,3X)
```

IPD034 DUMMY ARGUMENT EXPECTED

Explanation: A SUBROUTINE or FUNCTION statement has an argument that is invalid - a constant or an expression, for example. (This also applies to the ENTRY statement in FORTRAN G and H)

IPD036 ARRAY DIMENSIONS EXPECTED

Explanation: A DIMENSION statement gave either partial or no dimension information for one of its array names, or invalid characters were between the array name and the dimension information.

IPD038 / EXPECTED

Explanation: The first name in a NAMELIST statement was not preceded by a slash, or a name or list of data preceded by a slash was not followed by a slash. (COMMON, SUBROUTINE, FUNCTION, NAMELIST, DATA, INTEGER, REAL, LOGICAL, and COMPLEX statements in FORTRAN G and H only)

IPD040 INVALID DATA TYPE

Explanation: Either the type of a datum in a type-statement did not agree with the type declared by the statement, a datum was missing, or, in an IMPLICIT statement, the type specification was invalid (e.g., IMPLICIT REEL (D)). (Hexadecimal and both types of literal data are allowed in all type statements.) (FORTRAN G and H only)

IPD042 STATEMENT NUMBER EXPECTED

Explanation: A FORMAT statement was not numbered or a form other than a statement number was encountered after the GO TO in an unconditional GO TO statement. (In addition, in FORTRAN G and H, the invalid form could be in an ASSIGN or a CALL statement.)

IPD044 'TO' EXPECTED

Explanation: An ASSIGN statement did not have TO in the required place. (FORTRAN G and H only)

IPD046 ARGUMENT EXPECTED

Explanation: A CALL statement has an invalid or missing argument in its argument list.

IPD048 DATA LIST EXPECTED

Explanation: A DATA statement contains no data list or the data list is separated from the variable list by invalid characters. (FORTRAN G and H only)

IPD050 RELATIONAL OPERATOR EXPECTED

Explanation: In a logical expression, an arithmetic expression was not followed by a relational operator (FORTRAN G and H only)

IPD052 . EXPECTED

Explanation: In a statement with a relatively rigid form, such as EQUIVALENCE or DEFINE FILE, a comma was absent or preceded by invalid characters.

IPD054 OPERAND EXPECTED IN ARITH EXPRESSION

Explanation: An arithmetic operator was not followed by a valid arithmetic operand, or two operators occur together (as in A * -B).

IPD056 OPERAND EXPECTED IN LOGICAL EXPRESSION

Explanation: A logical operator was not followed by a logical operand, or a logical operand was missing. An invalid logical operator, such as ".NOT..NOT.", also causes this message. (FORTRAN G and H only)

IPD058 I/O LIST ITEM EXPECTED

Explanation: A variable name did not follow a comma in the list of a READ or WRITE statement (or in FORTRAN G and H, a PUNCH or PRINT statement).

IPD060 ' EXPECTED

Explanation: A FIND statement did not contain an apostrophe to separate the data set reference number from the expression describing the record to be found. This message is issued only if the omission of the apostrophe leaves a valid data set reference number. For example, "FIND (10 100)" would not cause this message because 10100 is not a valid data set reference number.

IPD062 INCORRECT PARAMETER - MUST BE E, L, OR U

Explanation: The data set control character in a DEFINE FILE statement was not an E, L, or U.

IPD064 DEBUG PARAMETER EXPECTED

Explanation: A valid parameter did not follow a comma after a valid parameter in a DEBUG statement. (FORTRAN G only)

IPD066 SUBSCRIPT EXPECTED

Explanation: A subscript was missing, not in one of the valid forms, or separated from the preceding comma or left parenthesis by invalid characters. (FORTRAN E only)

IPD068 TOO MANY LEVELS OF PARENTHESES

Explanation: In a FORMAT statement, group repeat specifications were too deeply nested. FORTRAN E does not allow a group repeat specification within a group repeat specification. FORTRAN G and H do allow this, but allow no further nesting of group repeat specifications.

IPD072 INTEGER EXPECTED

Explanation: A numeric constant that was not an integer was found where an integer is required.

IPD074 COMPLEX NUMBER INVALID

Explanation: The two parts of a complex constant did not agree in length, or one or both parts was not a real constant. (FORTRAN G and H only)

IPD076 DELIMITER MISSING OR INVALID FORMAT CODE

Explanation: In a FORMAT statement, either a format code was invalid, a delimiter (such as the required comma or slashes between two literal format codes) was missing, or the right parenthesis at the end of the format was missing.

IPD078 VARIABLE LIST EXPECTED

Explanation: In a DATA statement, a variable name (or list of variable names) did not occur (1) as the first item of the DATA statement, or (2) after a comma following a list of constants enclosed in slashes. This message is also issued when there are invalid characters between the DATA keyword or the comma and the list of variables. (FORTRAN G and H only)

IPD080 . EXPECTED IN FORMAT CODE

Explanation: There was no period as required in the "w.d" following the D, E, or F format code in a FORMAT statement.

IPD082 NAME TOO LONG

Explanation: A symbolic name contains more than six characters, a keyword of seven or more letters at the beginning of a statement is misspelled, or a misspelled keyword follows the "IF(logical expression)" part of a logical IF statement. A missing delimiter may cause this message to be issued. Each of the following statements would produce this message:

```
38 CONTINUE
SUBROUTINE X(ARG1 ARG2
```

IPD084 STATEMENT NUMBER INVALID

Explanation: The statement number field (positions 1-5 of the initial line of a statement) was zero or contained at least one character that was neither a digit nor a blank. Within a statement, this message is issued if a statement number is zero or contains too many digits.

IPD086 H-LITERAL INCOMPLETE

Explanation: The number of characters in the statement after the H was smaller than the count before the H in the H-literal.

IPD088 FIELD WIDTH NOT IN RANGE 1-255

Explanation: The field width specified in a format code was not in the required range.

IPD090 LITERAL EXCEEDS 255 CHARACTERS

Explanation: The number of characters between the apostrophes of a literal exceeds 255. In determining the number of characters enclosed, two adjacent apostrophes within the outermost apostrophes are counted as one character.

IPD092 STATEMENT ANALYSIS EXCEEDS TABLE LIMITS

Explanation: The statement was so complicated that the syntax checker ran out of space in its table. The statement cannot be checked by the syntax checker.

IPD094 END REQUIRES BLANK LABEL & CONTIN FIELDS

Explanation: Positions 1-6 of an END statement must be blank, and an END statement cannot have continuation lines.

IPD096 INVALID OR EXCESS SOURCE CHARACTERS

Explanation: The statement being checked completely satisfies the definition for that type of statement at some point before the last nonblank character in the statement. This may occur if a delimiter has been left out between elements of a statement.

IPD098 INVALID RANGE IN IMPLICIT STATEMENT

Explanation: An IMPLICIT statement contains a range of characters in which the last character of the range alphabetically precedes the first character of the range. Note that \$ follows Z in the alphabet. A character that is not alphabetic will also cause this message to be issued. (FORTRAN G and H only)

IPD100 FIRST LINE IS A CONTINUATION

Explanation: The first line scanned did not have C in position one, nor did it have a blank or zero in position six.

IPD102 COMMENT LINE WITHIN STATEMENT

Explanation: A comment line was found between an initial line and a continuation line or between two continuation lines.

IPD104 TOO MANY CONTINUATION LINES

Explanation: The statement consisted of an initial line followed by more than 19 continuation lines.

IPD106 TOO MANY DECIMAL PLACES FOR FIELD WIDTH

Explanation: In a D, E, F, or G format code, the fractional portion ("d" of "w.d") exceeded the total field width ("w" of "w.d"). (G in FORTRAN G and H only)

IPD108 DECIMAL PLACES MUST BE SPECIFIED

Explanation: In a D, E, or F format code, fractional portion ("d" of "w.d") was missing or was separated from the decimal point by invalid characters.

IPD110) REQUIRED FOR IMPLIED DO

Explanation: In the input or output list of an I/O statement, a right parenthesis was not the first non-blank character after the parameters of an implied DO.

IPD112 DO VARIABLE CANNOT BE SUBSCRIPTED

Explanation: In an I/O list, a subscripted variable was used as the DO variable within an implied DO loop.

IPD114 DEBUG FACILITY NOT SUPPORTED

Explanation: One of the Debug Facility statements appeared in a FORTRAN H program. These statements are allowed only in FORTRAN G. They are diagnosed when FORTRAN H is being checked since FORTRAN G and H use the same syntax table. If FORTRAN E is being checked, these statements will be diagnosed as unrecognizable or as beginning with too long a name.

IPD116 EXPONENT MISSING OR INVALID

Explanation: The characters after the D or E in a double precision or real constant do not constitute a valid exponent.

IPD118 REAL CONSTANT MUST HAVE AT LEAST 1 DIGIT

Explanation: In a place where a numeric constant might be written, a decimal point was found followed by an E or D exponent, either side of the decimal point.

IPD120 INTEGER TOO LARGE

Explanation: The magnitude of an integer constant exceeds 2147483647.

IPD122 CLOSING ' EXPECTED

Explanation: The end of the statement was reached without finding the closing apostrophe for a literal constant or literal format code. The opening apostrophe of the literal is the source character identified in the error message. (Only the literal format code is allowed in FORTRAN E.)

IPD124 DATA ILLEGAL FOR DUMMY ARRAY

Explanation: An array was dimensioned in a REAL, INTEGER, LOGICAL or COMPLEX type-statement. One of the dimensions was a variable name, making the array a dummy array, but a slash, indicating the start of a list of data, was then encountered. Dummy arrays cannot be assigned initial data values.

IPD126 REAL NUMBER EXPECTED

Explanation: A numeric datum of a type other than real was written in a REAL type-statement. (FORTRAN G and H only)

IPD128 INVALID CHARACTERS AFTER STOP OR PAUSE

Explanation: Characters other than digits, or too many digits, follow the STOP or PAUSE in a STOP or PAUSE statement. (In FORTRAN G and H, a literal is also permitted to follow PAUSE and will not receive this diagnostic.)

IPD130 REAL NUMBER OUTSIDE OF ALLOWABLE RANGE

Explanation: The magnitude of a real number, taking the value of the exponent into account, is outside the range 1.0E-79 to 9. (any fraction) E+75.

IPD132 FORMAT STMT NO. OR ARRAY NAME EXPECTED

Explanation: In a PRINT, PUNCH, or READ statement, no reference is made to a FORMAT statement or to an array containing a FORMAT. (FORTRAN G and H only)

IPD134 MISPLACED LENGTH SPECIFICATION PRECEDES

Explanation: A FUNCTION statement in which the FUNCTION is preceded by REAL, INTEGER, COMPLEX, or LOGICAL has a length specification between the type and FUNCTION. The length specification is invalid in that position and must be moved to the end of the function name.

IPD138 ARITH EXP EXPECTED AFTER RELATIONAL OP

Explanation: A relational operator in a logical expression was not followed by a valid arithmetic expression. (FORTRAN G and H only)

IPD140 INVALID COMMA IN DO

Explanation: An invalid comma was found after the statement number in a DO statement.

IPD142 = EXPECTED

Explanation: In a DO statement in which the statement number was followed by an invalid comma, the equal sign expected after the DO variable was not found.

IPD144 LITERAL CONTAINS NO CHARACTERS

Explanation: The closing apostrophe of a literal constant or literal format code occurred immediately to the right of the opening apostrophe. (Only the literal format code is allowed in FORTRAN E.)

IPD146 INVALID IF AFTER LOGICAL IF

Explanation: A statement after the "IF (logical expression)" part of a logical IF statement was recognized as an IF statement, but was not a valid arithmetic IF statement as required. (FORTRAN G and H only)

IPD148 INVALID DECIMAL POINT

Explanation: A real constant contains an extraneous decimal point.

IPD158 TOO MANY SUBSCRIPTS PRECEDE

Explanation: A list of eight or more (four or more in FORTRAN E) expressions within parentheses has been found following a symbolic name on the left side of the equal sign in an assignment statement.

IPD160 END TOO FAR ON LINE

Explanation: The free-form END statement contains more than 66 characters (not including trailing blanks). ("Free-form" refers to the free-form source input option of the Code and Go FORTRAN processor).

Note: If other syntax errors are detected in checking the statement, this message will not be sent; instead, messages applicable to the other errors will be sent.

The following functions, restrictions, and assignments are determined by the central installation when the system is generated.

ADDITIONAL COMMANDS AND SUBCOMMANDS

The installation may add commands and subcommands to the system by providing the routines to process them.

COMMAND ALIASES

The installation may assign alternate verbs (aliases) for the CRJE commands and subcommands. Duplication of aliases is allowed between modes but not within the same mode; i.e., the same alias may be used for a command and a subcommand, but it cannot be used for two commands (if in command mode) nor for two subcommands (if in edit mode). Either the CRJE name or the installation alias is recognized when entered from a terminal.

EXIT ROUTINES

Routines may be provided by the installation to check the accounting information on LOGON commands, to check JCL statements of jobs submitted for batch processing, and to obtain accounting information when a user logs off the system. An installation routine may reject a LOGON command and may terminate a job submission.

SYNTAX CHECKERS

The installation selects what syntax checkers, if any, are provided in the system and the kind of checking performed (i.e., level of checking or language level supported).

NUMBER OF LINES PER SYNTAX SCAN

The installation can impose a limit on the number of lines one statement can span and still be scanned as a complete statement by the syntax checker.

USERID/PASSWORD

The installation assigns userids and passwords to authorized terminal users.

CRJE SYSOUT CLASS

The system output class used for remote job output to be returned to terminal users is assigned by the installation.

NUMBER OF LINES PER OUTPUT GROUP

The installation specifies how many lines of output are sent to a terminal before allowing the terminal user to interrupt the output. This only applies to terminals without a special interrupt feature.

MAXIMUM NUMBER OF JOBS

The maximum number of jobs that can reside in the central system at one time is determined by the installation. When this maximum is reached, no more jobs are accepted until some of the existing jobs are canceled or their output is returned.

MAXIMUM NUMBER OF MESSAGES

The installation determines the number of messages that can be maintained by the system at any one time. This includes messages waiting for delivery at LOGON time and messages currently being processed.

ROUTING CODES FOR MULTIPLE CONSOLES

If the central system supports multiple consoles, the installation specifies a routing code for each console. A user may direct a message to an operator at a particular console by specifying the routing code for that console.

ON-LINE TERMINAL TEST

The installation determines whether or not the BTAM On-Line Terminal Test facility is provided. This facility provides tests that can be used by the terminal user as a start-up procedure or by the customer engineer for terminal checkout and diagnosis of terminal failure.

SESSION MANAGEMENT COMMANDS

<u>Command</u>	<u>Function</u>
LOGON	To identify the user and initiate his session.
LOGOFF	To terminate a session.

DATA MANAGEMENT COMMANDS

GENERAL

<u>Command</u>	<u>Function</u>
DELETE	To scratch an OS data set or to remove a CRJE data set from the user's library.
EDIT	To initiate creating or updating operations.

EDIT SUBCOMMANDS

<u>Subcommand</u>	<u>Abbreviation</u>	<u>Function</u>
INPUT	I	To insert and/or replace lines in the active data set.
DELETE	D	To remove lines in the active data set.
Implicit		To enter or delete lines in the active data set.
CHANGE	C	To replace character strings within lines of the active data set.
MERGE	M	To combine another data set with the active data set or to copy lines from one place to another within the active data set.
RENUMBER	REN	To reassign line numbers to the lines in the active data set.
LIST	L	To display lines of the active data set.
SCAN	SC	To request a syntax analysis of PL/I or FORTRAN source language statements in the active data set.
SAVE	S	To store the active data set in the user's library.
END		To terminate creating and updating operations and to delete the active data set.

JOB PROCESSING COMMANDS

<u>Command</u>	<u>Function</u>
SUBMIT	To enter a job into the OS job input stream. (Can also be used as an EDIT subcommand; it can be abbreviated -SUB- when used as a subcommand.)
OUTPUT	To request CRJE SYSOUT output of a conversationally-submitted job.
CONTINUE	To resume output listing that was previously interrupted.
CANCEL	To remove a job from the CRJE system and to delete any CRJE SYSOUT output of that job. (Can be used as an EDIT subcommand; it may also be abbreviated -CA- when used as a subcommand.)

STATUS INFORMATION COMMANDS

<u>Command</u>	<u>Function</u>
LISTLIB	To obtain the name and characteristics of every CRJE data set in the user's library.
LISTDS	To obtain information about a particular CRJE data set in the user's library.
STATUS	To obtain information about jobs the user has submitted.

MESSAGE COMMANDS

<u>Command</u>	<u>Function</u>
SEND	To send a message to the central operator or to another terminal user. (Can also be an EDIT subcommand.)
LISTBC	To request the broadcast messages.

TABSET COMMAND

<u>Command</u>	<u>Function</u>
TABSET	To indicate the tab settings at the terminal. This command affects all input and output and can be either a command or an EDIT subcommand. (Can only be abbreviated - TAB - as a subcommand.)

EXEC COMMAND

<u>Command</u>	<u>Function</u>
EXEC	To execute a sequence of commands contained in a CRJE data set.

Active area: A work area in direct access storage used for creating and manipulating data sets. This area is shared by all users.

Active data set: The data set that a user creates, updates, or lists in his active area.

Alias: An alternate name that a user may assign to a terminal command or subcommand.

Attribute: A characteristic of a data set.

Background processing: The automatic execution of lower-priority programs when higher-priority programs are not using the system resources.

Batch Processing: The processing of a stream of jobs by an operating system, controlled by statements supplied by the job originator prior to the commencement of processing.

Broadcast message: A message generated by the central operator that is of interest to, and sent to, all active users. It may notify the users of system closedown time, inactive devices, etc.

Cataloged data set: A data set that is represented in an index or hierarchy of indexes, which provides the means for locating it.

Closedown: Termination of computer processing and all terminal operations.

Command mode: The mode of operation in which the major commands, as opposed to the EDIT subcommands, are entered at the terminal to perform various functions (e.g., receiving job output). As long as a session is active and there is no active data set, you are operating in command mode.

Command prompt: A signal from the system indicating that it is ready for the user to enter a command (or a subcommand in edit mode).

Communication line: The medium over which data signals are transmitted.

Content attribute: The attribute that indicates the type of records that are contained in the data set (e.g., PL/I, FORTRAN).

Conversational processing: On-line processing that consists of a dialog between the terminal user and the central system.

CRJE data set: A data set that is created via the commands of CRJE and is stored in a user's library.

Data set name (dsname) (1) CRJE: A 1-8 character name that is assigned to a CRJE data set. The first character must be alphabetic. The name may contain the alphabetic characters A-Z, 0-9, @, #, and \$. It corresponds to a member name in the user's library. (2) OS: The name of an OS data set. It must be enclosed in quotes when specified on a CRJE command.

Edit mode: The mode in which the active data set is created by using EDIT subcommands and/or is manipulated-modified, updated, listed-by using EDIT subcommands.

Input line: A line of data entered from the terminal that becomes a record of the active data set.

Input mode: The mode in which input lines are entered that become records of the active data set.

Installation: The user's own particular configuration of devices, terminals, central processing unit, etc. All of the functions, restrictions, and assignments are determined by the central installation when the system is generated.

Key: A three-character identifier assigned to a CRJE data set that provides protection from unauthorized access.

Library: see "user library."

Line: see "communication line" or "input line."

Line increment attribute: The attribute that indicates the difference between the line numbers in a data set.

Line number: The number associated with each record, or input line, which enables you to refer to specific lines when requesting a listing or a modification.

Line number prompt: A signal from the system that it is ready to accept lines of input for the active data set. After the EDIT command or the INPUT subcommand has been entered, the system responds by printing the line number of the line to be entered. Prompting begins with line number 10 and continues in increments of 10 (unless otherwise specified).

MFT: Multiprogramming with a fixed number of tasks.

Multiprogramming: The programming method that permits a computer to perform several tasks simultaneously where input/output and computing may be overlapped.

MVT: Multiprogramming with a variable number of tasks.

Non-CRJE data set: A data set that was not created using CRJE (e.g., a data set that was created with utilities).

Null line: A line generated by entering the end-of-line sequence without entering any characters. It is used before logging on and when switching from edit mode to input mode, and conversely.

On-line: Pertaining to devices in direct communication with a computer.

On-line terminal test: A test that can be used by a terminal operator as a start-up procedure or by a customer engineer for terminal check-out and diagnosis of terminal failure. It operates on-line concurrently with the user's problem program(s) and in no way impacts user operation except for the line time required to perform the test's function on the selected line.

OS data set: A data set that is created via background processing using OS data management facilities. For an OS data set to be accessible by CRJE, it must be named, cataloged, resident on a direct-access storage device, and have either sequential or partitioned organization.

Password: A 1-8 character code associated with the userid which identifies you as an authorized system user at logon time. (The first character of the password must be alphabetic.)

Prompt: A signal from the system indicating that it is ready for the user to enter either a command, a subcommand, a line of input, a userid, a password, or a dsname. (See "underscore prompt" and "line number prompt.")

Protection key: See "key."

Record: A series of characters treated as a unit. Each line of input entered from a terminal is considered a record.

Sequence attribute: The attribute that indicates whether line numbers are kept in positions 73-80 of each record.

Session: The period of time that a user is active at a terminal. It is initiated by LOGON; it is terminated by either LOGOFF, another user's LOGON, a line error, or central closedown.

Start-up: Initiation of computer processing or resumption of it from a point of temporary termination.

Switched communication line: A communication line that requires dialing in order to establish a connection with the central system.

Syntax checker: A facility that analyzes statements in a language to be sure that the structure of the individual statements is correct. (Applicable only to FORTRAN and PL/I in CRJE.)

SYSOUT class: see "system output class."

System output class: The class specified in a Data Definition (DD) statement, which indicates the destination of any output produced by a job.

Telecommunications: Data transmission between a computing system and remotely located devices via a unit that performs the necessary format conversion and controls the rate of transmission.

Terminal: Any device capable of sending and/or receiving information over a communication line.

Turnaround time: The elapsed time between submission of a job to a computing center and the return of results; the time required to submit a job and to receive its output.

Underscore prompt: A signal from the system that it is ready to accept either a command or an EDIT subcommand. The prompt consists of an underscore (_) followed by a backspace and the unlocking of the terminal keyboard.

- abbreviating EDIT subcommands 17,41
- abnormal session termination 14,52
- accounting information
 - on the LOGON command 12,45
- active area
 - contents of 9
 - copying CRJE data set into 17
 - copying OS data set into 34
 - CRJE data set in 21
 - definition 15,159
 - DSLISL in 31
- active data set
 - command description 61-63
 - creating 15,46-51
 - definition 159
 - deleting 63-64
 - deleting lines in 54-55
 - entering lines in 52
 - in DSLISL (data set list) 30-31
 - listing 59-60
 - merging data within 56-58
 - NOSEQ attribute 59
 - OS data sets as 34,51-52
 - renumbering lines in 22-23,58-59
 - saving a data set 20
 - saving an updated data set 22-23
 - submitting 29-30,64-65
 - syntax checking statements in 23,60-61
 - updating 21-22,52-59
 - when logging off 14
 - with INPUT subcommand 53-54
- ACTIVE data set name 14
- aliases for commands and subcommands 151
- attribute 15-16
 - assigned to OS data set 34,52
 - content 15-16
 - data set 15-16
 - defined 15,159
 - line increment 16
 - (see also sequence attribute)
 - sequence 16
 - type (see content)
- automatic scan facility 24-27
 - obtaining via EDIT command 48
 - obtaining via SCAN subcommand 60-61
- batch processing 64-65,159
- broadcast messages 12-13,159
 - with LISTBC 72
 - with LOGON 45
- CANCEL command and subcommand 65-66
 - format 157,158
 - function 154
- cancelling jobs 35
- cancelling a line
 - 1050 terminal 125
 - 2740 terminal 120
- 2741 terminal 115
- central operator
 - generating broadcast messages to 12-13,37
 - receiving messages from 12-13
 - sending messages to 37,72-73
- central processing unit (CPU) 9
- CHANGE subcommand 55-56
 - format 158
 - function 154
 - making corrections with 19
 - in FORTRAN 141
 - in PL/I 129
- CLIST (see command list)
- CLIST data set 76-77
- command aliases 151
- command format 41-42,157-158
- command functions 153-155
- command list (CLIST)
 - content attribute 16
 - correcting a 39
 - creating a 38
 - executing 38-39
 - in EDIT command 49
 - in EXEC command 77
 - interrupting output of 39
 - using a 38-39
- command mode 16-17
 - definition of 159
 - returning to, from edit mode 20,62-63
- command notation 42-44
- command prompt 159
- commands
 - CANCEL 65-66
 - format 157
 - function 154
 - CONTINUE 67-68
 - format 157
 - function 154
 - DELETE 63-64
 - format 157
 - function 153
 - EDIT 46-53
 - format 157
 - function 153
 - EXEC 76-78
 - format 157
 - function 155
 - job processing 64-68
 - function 154
 - LISTBC 73
 - format 157
 - function 155
 - LISTDS 69-70
 - format 157
 - function 155
 - LISTLIB 70-71
 - format 157
 - function 155
 - LOGOFF 46
 - format 157
 - function 153

- with another's logon 14
- LOGON 44-45
 - after automatic LOGOFF 14
 - format 157
 - function 153
- message commands 72-73,155
- OUTPUT 66-67
 - format 157
 - function 154
- SEND 72-73
 - format 157
 - function 155
- session management 44-46
 - function 153
- special 73-78
- STATUS 68-69
 - format 157
 - function 155
- status information 68-71
 - function 155
- SUBMIT 64-65
 - format 157
 - function 154
- TABSET 74-76
 - format 157
 - function 155
- commands, data management 46-64
- communication line 9,159
- content attribute (type attribute) 15-16,159
 - assigning to a data set 48-49
- CLIST 16
 - in EDIT command 49
 - using a 38-39
- DATA 16
- DSLISIT 16,49
- FORT 16
 - in EDIT command 49
 - in SCAN subcommand 60
- PL/I 16
 - in EDIT command 48
 - in SCAN subcommand 60
- TEXT 16
 - when creating a data set 51
 - when editing a data set 51
 - when obtaining data set status 36
- continuation of a command or subcommand 42
- CONTINUE 33,67-68
 - format 157
 - function 154
 - resuming output with 67
- correcting a CLIST 39
- correcting a line
 - 1050 terminal 125
 - 2740 terminal 120
 - 2741 terminal 115
- correcting syntax errors 60-61
 - user action with FORTRAN 141
 - user action with PL/I 129
- CPU (central processing unit) 9
- creating a CLIST 38
- creating a CRJE data set 18-19,50-51
 - (see also EDIT command)
- CRJE data activity 35
- CRJE data set
 - creating 18-19,50-51
 - definition of 159
 - deleting 33,63-64
 - editing 21-22,51
 - in DSLISIT 16
 - merging, with active data set 56-58
 - saving 20
 - submitting 28
 - updating 21-23
- CRJE instream procedures, restriction in MFT 65
- CRJE modes of operation 17
- CRJE SYSOUT class 152
- DASD (see direct access storage device)
- DATA content attribute 16
- Data Definition (DD) statement
 - for specifying input 28
 - for specifying output 29
- data management commands 46-64
 - DELETE 63-64
 - EDIT 46-52
 - functions of 153-154
- data set
 - active (see active data set)
 - creating a 18-20
 - CRJE
 - creating with edit command 50-51
 - deleting 33
 - editing 21-23,51
 - listing a 19-20
 - OS, editing 51-52
 - saving 20
 - updated 22-23
 - data set attribute 15-16
 - data set list (DSLISIT)
 - content attribute 16
 - in EDIT command 49
 - submitting 30-32,64
 - data set name, restriction 52
 - data set status 36-37
 - DD statement (see Data Definition statement)
 - DELETE command 63-64
 - compared with DELETE subcommand 23
 - format 157
 - function 153
 - DELETE subcommand 23,54-55
 - format 158
 - function 154
 - deleting
 - active data set 63-64
 - CRJE data set 23,63-64
 - OS data set 63-64
 - deleting lines in a data set 21,54-55
 - delimiters 19,41
 - on CHANGE subcommand 55-56
 - on Implicit subcommand 55
 - diagnostic message
 - with syntax checking 23
 - (see also terminal messages)
 - direct-access storage device (DASD) 29,52
 - DSLISIT (see data set list)
 - DSLISIT data set 49,64
 - nesting restriction 16
 - submitting 30-31

- lsname (data set name)
 - definition 43,159
 - format in terminal commands 43
 - when creating a data set 18
 - when obtaining data set status 36
- EDIT command 46-53
 - to create an active data set 16-17
 - to create a CRJE data set 18-19,50
 - to edit a CRJE data set 50-51
 - to edit an OS data set 51-52
 - format 157
 - function 153
- edit mode 17,49
 - definition 159
 - ending 20
 - entering 17
- EDIT subcommands 52-63,154
 - abbreviating 17,41
 - CANCEL 65-66
 - CHANGE 55-56
 - DELETE 54-55
 - END 63
 - format 158
 - functions of 154
 - Implicit 55
 - INPUT 53-54
 - LIST 59
 - MERGE 56-58
 - RENUMBER 58
 - SAVE 61-63
 - SCAN 60-61
 - SEND 72-73
 - SUBMIT 64
 - TABSET 73-76
 - used with OS data sets 52
- editing an active data set 16-17
 - after abnormal session termination 52
- editing a CRJE data set 21-23
 - after abnormal session termination 52
 - with EDIT command 51
- editing an OS data set 51-52
 - after abnormal session termination 52
- end-of-line sequence
 - 1050 terminal 123
 - 2740 terminal 119
 - 2741 terminal 115
- END subcommand 63
 - changing modes 16-17
 - ending edit mode 20,49
 - format 158
 - function 154
 - when logging off 14
 - when resuming CLIST execution 39
- entering edit mode 17
- entering input mode 17,52
- entering a line
 - 1050 terminal 124-125
 - 2740 terminal 119-120
 - 2741 terminal 115
- error conditions
 - in PL/I statements 129
 - 1050 terminal 126-127
 - 2740 terminal 120-121
 - 2741 terminal 116
- error messages 23-24
 - FORTRAN 141-150
 - PL/I syntax checker 129-140
- errors, correcting 24
- establishing a connection (dial lines) 11-12
- EXEC command 76-78
 - format 157
 - function 155
- executing a command list (CLIST) 38-39
- exit routines 151
- format, command 41-42
- format illustration conventions 42-43
- FORT content attribute 16
 - in EDIT command 49
 - in SCAN subcommand 60
- FORTRAN source statements 16,25-26
 - continuing 16
- FORTRAN Syntax Checker 24,49
 - (see also SCAN subcommand)
 - messages 141-150
- functions of terminal commands 153-155
- gaining access to the system 11-14
- identifying jobs 28
- Implicit subcommand 19,55
 - to delete a line 21
 - format 158
 - function 154
- increment 17,44
 - on the INPUT subcommand 53
 - on the RENUMBER subcommand 58
- initiating a session 11-13
- input, specifying 28,64
- input line 52-53,159
- input mode 17-18,49
 - entering via EDIT command 17,160
 - entering via INPUT subcommand 21,52-54
 - inserting lines in 21
 - terminating 54
- INPUT subcommand 21,52-54
 - format 158
 - function 154
- inserting lines in a data set 21,53-54
- installation variables 151-152
- instream procedures, MFT support restriction 28
- interrupting output
 - during CLIST 39
 - while listing active data sets 59
 - while listing broadcast messages 73
 - while listing OS data sets 34
 - via Interrupt feature 32,67
 - 1050 terminal 125-126
 - 2740 terminal 120
 - 2741 terminal 116

JCL (see Job Control Language)

- job 28,64
 - cancelling 35
 - identifying 28
 - maximum number of 152
 - submitting 28-32
- Job Control Language (JCL) 16,65
 - as input for job submission 28,66
- job input 27
 - specifying 28,64
- job management messages 29,66-67
- job output
 - discontinuing transmission of 32-33
 - specifying 29
- job processing 28-35
- job processing commands 64-68
 - CANCEL 65-66
 - CONTINUE 67-68
 - functions of 154
 - OUTPUT 66-67
 - SUBMIT 64-65
- job status 36
- JOBFAIL, job name restriction 28
- jobname 28,44
 - restriction 28
- key, protection 21
 - definition 160
 - terminology 43,44
- keyboard
 - 1050 terminal 124
 - 2740 terminal 121
 - 2741 terminal 117
- line (see communication line or input line)
- line increment attribute
 - assigning to a new data set 50
 - definition of 16,160
 - when inserting lines 21
 - when merging lines 22
 - when renumbering lines 22,58-59
- line number (linenum) 15
 - assigned to OS data set 52
 - assigned via RENUMBER 58-59
 - creating a data set 18-19,49-50
 - deleting a line 21-22
 - listing a data set 19-20
 - sequence attribute 16
 - terminology 44
- line number prompt
 - definition of 160
 - in INPUT command 17,53
 - in TABSET command, allowing for 74
 - when creating a data set 50
- line sequence number (see line number prompt)
- linenum (see line number)
- lines
 - deleting 21
 - inserting 21
 - moving 22
 - number per output group 152
 - number per syntax scan 151
 - renumbering 22
- LIST subcommand 19-20,59-60
 - with OS data set 34
 - format 158
 - function 154
- LISTBC command 73
 - format 157
 - function 155
- LISTDS command 69-70
 - format 157
 - function 155
- listing a data set 19-20
- LISTLIB command 70-71
 - format 157
 - function 155
- LOGOFF command 46
 - format 157
 - function 153
 - with another's LOGON 14
- LOGON command 12-13,44-45
 - format 157
 - function 153
 - logging off via another user's LOGON 14
- making corrections 19
 - to FORTRAN statements 60,141
 - to PL/I statements 60,129
- maximum number of jobs 152
- maximum number of messages 152
- MERGE subcommand 56-58
 - format 158
 - function 154
- message codes 14
 - for LOGON 45
 - for terminal messages 79
- message commands
 - functions of 155
 - LISTBC 73
 - SEND 72-73
- messages
 - central operator 37
 - delivery 13
 - diagnostic 24
 - error
 - FORTRAN 141-150
 - PL/I 129-140
 - format of 129-140,141-150
 - job management of 141-150
 - maximum number of 152
 - PL/I syntax checker 129-140
 - sending 37-38
 - syntax checker
 - FORTRAN 141-150
 - PL/I 129-140
 - terminal
 - numerical listing of 82-113
 - subject listing 79-82
 - to terminal user 37-38
- modes of operation 16-17
- moving lines in the active data set 22,56-58
- multiple consoles 72,152

NOSEQ attribute (see sequence attribute)
 in active data set 59

null line
 before logging on 12
 definition 160
 to switch modes 49-50, 53

number of lines per output group 152

number of lines per syntax scan 151

numbering of lines 15

numerical listing
 FORTRAN syntax checker messages 141-150
 PL/I syntax checker messages 130-140
 of terminal messages 82-113

obtaining output 32-35

obtaining status information 36-37

on-line terminal test (BTAM) 152
 1050 terminal 127
 2740 terminal 121
 2741 terminal 117

operand field of a command 11, 41

operation field of a command 11, 41

operational modes 16-17

OS data management facilities 9

OS data set 34, 47-48
 as part of output 29
 as the active data set 51-52
 definition 160
 deleting 34, 63-64
 line number assignment 52
 obtaining a listing of 34
 record size 51
 submitting 29-30

OS job management messages 29, 66-67

OS job management routines 9, 64

OS procedure library 28

output
 obtaining 32-35
 specifying 29
 SYSOUT 32-34

OUTPUT command 66-67
 format 157
 function 154
 used with CONTINUE command 68

password 44, 160
 assigning 44, 151
 incorrectly specified 12

PL/I content attribute 16
 in EDIT command 48
 in SCAN subcommand 60

PL/I statements 24-25
 attribute for 15, 48
 continuing 25
 syntax checking 23-27

PL/I Syntax Checker 23
 error messages 129-140

procedures
 instream MFT support restriction 28

processing jobs 28-35

prompts
 command (see command prompt)
 line number (see line number prompt)
 underscore (see underscore prompt)

protection key (see key)

readying of terminal 11-12

record, 80-character 15, 20

record format 15

record size, of OS data set 51

RENUMBER subcommand 58-59
 format 158
 function 154

renumbering lines in the active data set 22-23, 58-59

requesting status information 69-70
 of data sets 36-37
 of jobs 36

routines, exit 151

routing codes for multiple consoles 72, 152

sample programs
 FORTRAN 25-26
 PL/I 24-25

SAVE subcommand 61-63
 format 158
 function 154
 with data sets 22-23, 51-52

saving a data set 20

saving an OS data set 52

saving the updated data set 22-23

SCAN subcommand 26-27, 60-61
 format 158
 function 154

SEND command 72-73
 format 157, 158
 function 155

sending messages 37-38

SEQ attribute (see sequence attribute)

sequence attribute 16, 161
 assigning to a data set 47
 NOSEQ 16
 SEQ 16
 when creating a data set 50
 when editing a data set 51
 when obtaining a data set 36

session
 beginning 12-13
 definition 161
 ending 13-15

session initiation 12-13

session management commands 44-46
 function 153

session, terminating 13-15
 abnormal 14, 52
 normal 46

special commands 73-78
 functions of 155

specifying input 28

specifying output 29

statements
 FORTRAN 25-26
 PL/I 24-25
 syntax checking 23-27

STATUS command 68-69
 format 157
 function 155

status of data sets 36-37

status of jobs 36

status information commands 68-71
 function 155

status information, obtaining 36-37

- subcommands
 - CHANGE 55-56
 - format 158
 - function 154
 - DELETE 54-55
 - format 158
 - function 154
 - of EDIT 52-63, 154
 - END 63
 - format 158
 - function 154
 - Implicit 55
 - format 158
 - function 154
 - INPUT 52-54
 - format 158
 - function 154
 - LIST 59-60
 - format 158
 - function 154
 - MERGE 56-58
 - format 158
 - function 154
 - RENUMBER 58-59
 - format 158
 - function 154
 - SAVE 61-63
 - format 158
 - function 154
 - SCAN 26-27, 60-61
 - format 158
 - function 154
 - SUBMIT 64-65
 - format 158
 - function 154
- subject listing of terminal messages 79-82
- SUBMIT command 64-65
 - format 157, 158
 - function 154
- submitting the active data set 29-30
- submitting a data set from a user's library 30
- submitting a data set list (DSL) 30-32, 64
- submitting jobs 28-32, 64
- switched line communication 45, 161
- syntax checker messages
 - FORTRAN 141-150
 - PL/I 129-140
- syntax checkers 151
- syntax checking
 - active data set 23, 60-61
 - provided for system 151, 161
 - with FORTRAN 24, 141-150
 - with PL/I 23, 129-140
- syntax checking PL/I and FORTRAN statements 23-27
- SYSOUT (see system output class)
- SYSOUT output 32-34, 66-67
- system messages 29, 66
- system output class (SYSOUT) 9, 152
 - assigning 152
 - data sets in 66-67
 - deleting a data set in 32-34
 - deleting job output in 67
 - requesting output in 29
- TABSET command 74-76
 - format 157, 158
 - function 155
- terminal
 - operating procedures
 - 1050 terminal 123-129
 - 2740 terminal 119-121
 - 2741 terminal 115-117
 - types of terminals supported 11
 - terminal command language 5
 - format of 41-42, 157-158
 - function of 153-155
 - terminal commands and functions 153-155
 - terminal messages 79-113
 - format 79
 - numerical listing 82-113
 - subject listing 79-82
 - terminal operation
 - 1050 terminal 123-129
 - 2740 terminal 119-121
 - 2741 terminal 115-117
 - terminal test, on-line 152
 - 1050 terminal 127
 - 2740 terminal 121
 - 2741 terminal 117
 - terminal user, messages to 37-38
 - terminating a session 13-15
 - terminology used in command format 43-44
 - TEXT content attribute 16
 - timeout suppression feature (1050 terminal) 127
 - turning on terminal
 - 1050 terminal 123
 - 2740 terminal 119
 - 2741 terminal 115
 - type attribute (see content attribute)
- underscore prompt 12, 161
 - before logging on 45
 - responding with EDIT subcommands to 49-50
- updating facilities 46, 50-52
- user id 44-45, 161
 - assigning 151
 - format in terminal commands 44
 - incorrectly, specified 12
- user library 9
 - attributes of data set in 15
 - listing names of data set in 70-71
 - merging lines from 19, 56-58
 - saving a data set in 50, 61
 - submitting a data set 30
- using a command list (CLIST) 38-39
- variables, installation 151-156
- 72-character record 15, 20
- 80-character record 15, 20
- 1050 terminal 123-129
- 1052 printer-keyboard 124
- 2740 terminal 119-121
- 2741 terminal operation 115-117

READER'S COMMENTS

TITLE: IBM System/360 Operating System
Conversational Remote Job Entry
Terminal User's Guide

ORDER NO. GC30-2014-1

Your comments assist us in improving the usefulness of our publications; they are an important part of the input used in preparing updates to the publications. All comments and suggestions become the property of IBM.

Please do not use this form for technical questions about the system or for requests for additional publications; this only delays the response. Instead, direct your inquiries or requests to your IBM representative or to the IBM Branch Office serving your locality.

Corrections or clarifications needed:

Page *Comment*

Please include your name and address in the space below if you wish a reply.

Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

out along this line

fold

fold

FIRST CLASS
PERMIT NO. 33504
NEW YORK, N.Y.

BUSINESS REPLY MAIL
NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES



POSTAGE WILL BE PAID BY . . .

IBM CORPORATION
1271 Avenue of the Americas
New York, New York 10020

Attention: PUBLICATIONS

fold

fold



International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
[U.S.A. only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

IBM S/360 CRJE Terminal User's Guide (S/360) Printed in U.S.A. GC30-2014-1

Date of Publication: January 1972
Form of Publication: Revision, GC30-2014-1

80-Character Record Support

Specification Change

CRJE now supports 80-character records in the user library. Discussion is found in "Creating, Editing and Deleting CRJE Data Sets," in "Editing an OS Data Set," in the SAVE command discussion and in "Terminal Messages."

Terminal Messages

Specification Change

The section "Terminal Messages" contains additional error messages and problem determination changes.

Miscellaneous Changes

Maintenance: Programming and Documentation

Restriction on Job Name: Both the Job Processing section and the SUBMIT command description note that JOBFAIL must not be used as the name of a remotely submitted job.

Resend Error: Appendix C includes in the Error Conditions section a correction for a resend error.

Maintenance: Documentation Only

CRJE does not support in-stream procedures under MFT. This is discussed in "Job Processing" and in the SUBMIT command description.

The LIST subcommand description includes a correction for the NONUM operand.

The SUBMIT command description includes a note that CRJE does not support in-stream procedures under MFT.

CANCEL Command: The effect of the CANCEL command on the SYSOUT data set is clarified.

EXEC Command: The clarification of the EXEC command is included.

Editorial changes that have no technical significance are not noted here.

Specific changes to the text made as of this publishing date are indicated by a vertical bar to the left of the text. These bars will be deleted at any subsequent republication of the page affected.



International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
[U.S.A. only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]