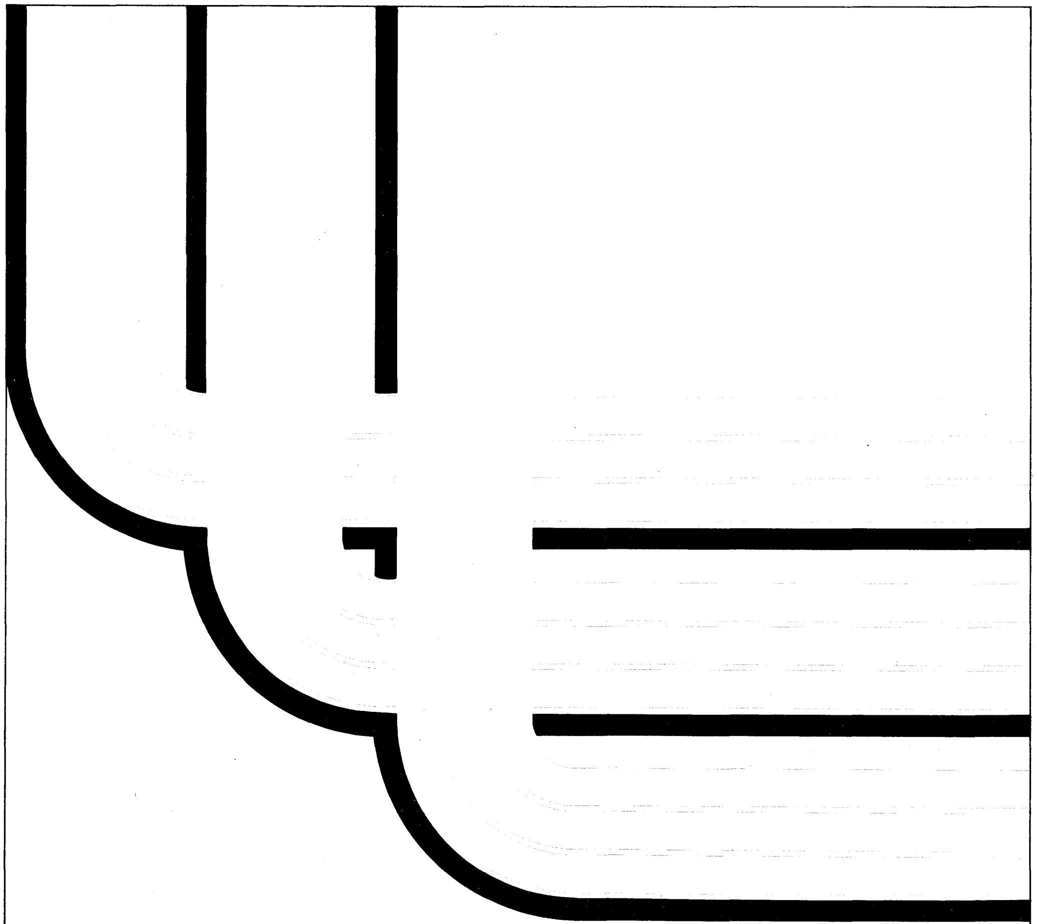
Application System/400

SC41-8083-02

Security Reference

Version 2



System and Application Support

SC41-8083-02

Application System/400

Security Reference

Version 2

Take Note!

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

Third Edition (November 1993)

This edition applies to the licensed program IBM Operating System/400 (Program 5738-SS1), Version 2 Release 3 Modification 0, and to all subsequent releases and modifications until otherwise indicated in new editions. This major revision makes obsolete SC41-8083-01. Make sure you are using the proper edition for the level of the product.

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

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

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

or you can fax your comments to:

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

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you or restricting your use of it.

Contents

Figures

Tables

Notices

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

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

This publication could contain technical inaccuracies or typographical errors.

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

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

Refer to the “Summary of Changes” on page xiii for a summary of changes made to the Operating System/400 * licensed program and how they are described in this publication.

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

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

Trademarks and Service Marks

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

Advanced Function Printing
Application System/400
AS/400
C/400
CallPath
COBOL/400
FORTRAN/400
IBM
Operating System/400
Operational Assistant
OS/400
PC Support/400
RM/COBOL-85
RPG/400
SQL/400
400

About This Manual

This manual provides information about planning, setting up, managing, and auditing security on your AS/400 system. It describes all the features of security on the system and discusses how security features relate to other aspects of the system, such as work management, backup and recovery, and application design.

The primary audience for this manual is the security administrator.

The information in the following chapters can help the application programmer and systems programmer understand the relationship between security and application and system design:

Chapter 5, “Resource Security”
Chapter 6, “Security and Work Management”
Chapter 7, “Designing Security”
Chapter 8, “Backup and Recovery”

Chapter 9, “Auditing Security on the AS/400 System” is intended for anyone who wants to perform a security audit of the system.

This manual assumes you are familiar with entering commands on the system. To use some of the examples in this book, you need to know how to:

- Edit and create a control language (CL) program.
- Install and use the Tips and Techniques (TAA) tools found in the QUSRTOOL library.
- Use a query tool, such as the Query/400 licensed program.

Throughout this manual are overview boxes intended to provide a quick reference to the feature being described:

Overview

Purpose: The task being described.

How To: What command or menu to use to accomplish the task.

Authority: Any special authority or object authority you need to do the task.

Journal Entry: What entry type, if any, is written to the security audit journal when you do the task.

Notes: Additional considerations.

This manual and the *Basic Security Guide*, SC41-0047, are replacements for the *Security Concepts and Planning* manual. The *Basic Security Guide* describes planning and setting up basic security on the system. It is intended for a system administrator who does not have a technical background.

This manual does not provide complete operational instructions for setting up security on your system. For a step-by-step example of setting up security, consult the *Basic Security Guide*.

This manual does not provide complete information about planning for OfficeVision/400 users and securing OfficeVision/400 objects. Planning for OfficeVision/400 users is described in *Systems Application Architecture* OfficeVision/400*: Planning For and Setting Up OfficeVision/400*, SC41-9626. Securing OfficeVision/400 objects is discussed in the *Office Services Concepts and Programmer's Guide*, SC41-9758.

This manual does not contain complete information about the application programming interfaces (APIs) that are available to access security information. APIs are described in the *System Programmer's Interface Reference*, SC41-8223.

This manual does not contain instructions for enabling C2 security on your system. The steps you must take to meet the U.S. Department of Defense requirements for C2 security are described in the *Guide to Enabling C2 Security*.

You may need to refer to other IBM manuals for more specific information about a particular topic. The *Publications Guide*, GC41-9678, provides information on all the manuals in the AS/400 library.

For a list of related publications, see the Bibliography.

Summary of Changes

Security Level 50: A fifth security level has been added to provide enhanced integrity protection and a more complete separation of the system state and the user state. The additional protection available with security level 50 has been designed to meet the requirements of C2 security, as defined by the United States government in the *Department of Defense Trusted Computer System Evaluation Criteria.*

A new system value has been added to enforce the separation of user and system state objects:

QALWUSRDMN The Allow User Domain Objects system value determines which libraries may contain user domain objects of type *USRSPC, *USRIDX, and *USRQ.

New Security Auditing Support: The security auditing capability of the system has been enhanced to provide additional options for auditing actions and the ability to audit successful access to objects. This support is provided by:

- New system values:

  QAUDCTL The Auditing Control system value determines what type of auditing is active on the system.

  QAUDENDACN The Auditing End Action system value determines what the system does if it is unable to write to the security audit journal.

  QAUDFRCLVL The Auditing Force Level system value determines how often the system writes security audit records to auxiliary storage.

  QCRTOBJAUD The Create Object Audit system value determines the default object auditing for new objects on the system.

- Additional choices for the QAUDLVL (Auditing Level) system value:

  ***JOBDTA** Audits actions that affect a job.

  ***OFCSRV** Audits changes to the system directory and mail actions.

  ***PGMADP** Audits the use of adopted authority.

  ***PRTDTA** Audits printing.

  ***SERVICE** Audits the use of service tools.

  ***SPLFDTA** Audits actions performed on spooled files.

  ***SYSMGT** Audits system management functions.

- A new *AUDIT special authority to specify which users can manage auditing on the system.

- New user profile parameters:

  AUDLVL The audit level parameter determines action auditing for the user.

  OBJAUD The object auditing parameter determines what object auditing is done for the user.

- A new library parameter, CRTOBJAUD, to specify the default object auditing for new objects in the library.

- A new object parameter, OBJAUD, specifies what auditing is done for an object.

- New commands:

  CHGUSRAUD The Change User Audit command is used to set the AUDLVL and OBJAUD parameters in a user profile.

  CHGOBJAUD The Change Object Auditing command is used to set the object auditing value for an object.

  CHGDLOAUD The Change DLO Auditing command is used to the set the object auditing value for a document library object.

- New and changed record layouts for the QAUDJRN field reference files to support the additional auditing capabilities.

National Language Support: As part of the new sorting capabilities for national language support, a sort sequence (SRTSEQ) parameter has been added to the user profile.

Chapter 1. Introduction

The Application System/400* family of systems covers a wide range of users. A small system might have three to five users, and a large system might have several hundred users. Some installations have all their workstations in a single, relatively secure, area. Others have widely distributed users, including users who connect by dialing in and indirect users connected through personal computers or system networks.

Security on the AS/400* system is flexible enough to meet the requirements of this wide range of users and situations. You need to understand the features and options available so that you can adapt them to your own security requirements. This chapter provides an overview of the security features on the system.

System security has three important objectives:

Confidentiality

- Protecting against disclosing information to unauthorized people.
- Restricting access to confidential information.
- Protecting against curious system users and outsiders.

Integrity

- Protecting against unauthorized changes to data.
- Restricting manipulation of data to authorized programs.
- Providing assurance that data is trustworthy.

Availability

- Preventing accidental changes or destruction of data.
- Protecting against attempts by outsiders to abuse or destroy system resources.

System security is often associated with external threats, such as hackers or business rivals. However, protection against system accidents by authorized system users is often the greatest benefit of a well-designed security system. In a system without good security features, pressing the wrong key might result in deleting important information. System security can prevent this type of accident.

The best security system functions cannot produce good results without good planning. Security that is set up in small pieces, without planning, can be confusing. It is difficult to maintain and to audit. Planning does not imply designing the security for every file, program, and device in advance. It does imply establishing an overall approach to security on the system and communicating that approach to application designers, programmers, and system users.

As you plan security on your system and decide how much security you need, consider these questions:

- Is there a company policy or standard that requires a certain level of security?
- Do the company auditors require some level of security?
- How important is your system and the data on it to your business?
- How important is the error protection provided by the security features?
- What are your company security requirements for the future?

Physical Security

Physical security includes protecting the system unit, system devices, and backup media from accidental or deliberate damage. Most measures you take to ensure the physical security of your system are external to the system. However, the system is equipped with a keylock that prevents unauthorized functions at the system unit.

Physical security is described in the *Basic Security Guide*.

Security Level

You can choose how much security you want the system to enforce by setting the security level (QSECURITY) system value. The system offers four levels of security:

Level 10: The system does not enforce any security.

Level 20: The system requires a user ID and password for sign-on. All users are given access to all objects.

Level 30: The system requires a user ID and password for sign-on. The security of resources is enforced.

Level 40: The system requires a user ID and password for sign-on. The security of resources is enforced. Additional integrity protection features are also enforced.

Level 50: The system requires a user ID and password for sign-on. The security of resources is enforced. Level 40 integrity protection and enhanced integrity protection are enforced. Security level 50 is intended for AS/400 systems with high security requirements, and it is designed to meet C2 security requirements.

The system security levels are described in Chapter 2.

System Values

System values allow you to customize many characteristics of your system. A group of system values are used to define system-wide security settings. For example, you can specify:

- How many sign-on attempts you allow at a device.
- Whether the system automatically signs off an inactive workstation.
- How often passwords need to be changed.

- The length and composition of passwords.

The system values that relate to security are described in Chapter 3.

User Profiles

Every system user has a user profile. At security level 10, the system automatically creates a profile when a user first signs on. At higher security levels, you must create a user profile before a user can sign on.

The user profile is a powerful and flexible tool. It controls what the user can do and customizes the way the system appears to the user. Following are descriptions of a few important security features of the user profile:

User class and special authority
The user class and special authority determine whether the user is allowed to perform system functions, such as creating user profiles or changing the jobs of other users.

Initial menu and initial program
The initial menu and program determine what the user sees after signing on the system. You can limit a user to a specific set of tasks by restricting the user to an initial menu.

Limit capabilities
The limit capabilities field in the user profile determines whether the user can enter commands and change the initial menu or initial program when signing on.

User profiles are discussed in Chapter 4.

Group Profiles

A group profile is a special type of user profile. You can use a group profile to define authority for a group of users, rather than giving authority to each user individually. A group profile can own objects on the system. You can also use a group profile as a pattern when creating individual user profiles by using the copy profile function.

"Planning Group Profiles" on page 7-10 discusses using group authority. "Group Ownership of Objects" on page 5-6 discusses what objects should be owned by group profiles. "Copying User Profiles" on page 4-19 describes how to copy a group profile to create an individual user profile.

Resource Security

Resource security on the system allows you to define who can use objects and how those objects can be used. The ability to access an object is called **authority**. You can specify detailed authorities, such as adding records or changing records. Or you can use the system-defined subsets of authorities: *ALL, *CHANGE, *USE, and *EXCLUDE.

Files, programs, and libraries are the most common objects requiring security protection, but you can specify authority foı any object on the system. Following are descriptions of the features of resource security:

Group profiles
A group of similar users can share the same authority to use objects.

Authorization lists
Objects with similar security needs can be grouped on one list; authority can be granted to the list rather than to the individual objects.

Object ownership
Every object on the system has an owner. Objects can be owned by an individual user profile or by a group profile. Proper assignment of object ownership helps you manage applications and delegate responsibility for the security of your information.

Library authority
You can put files and programs that have similar protection requirements into a library and restrict access to that library. This is often easier than restricting access to each individual object.

Object authority
In cases where restricting access to a library is not specific enough, you can restrict authority to access individual objects.

Public authority
For each object, you can define what kind of access is available for any system user who does not have any other authority to the object. Public authority is an effective means for securing information and provides good performance.

Adopted authority
Adopted authority adds the authority of a program owner to the authority of the user running the program Adopted authority is a useful tool when a user needs different authority for an object, depending on the situation.

Authority holder
An authority holder stores the authority information for a program-described database file. The authority infor mation remains, even when the file is deleted. Authority holders are commonly used when converting from the System/36, because System/36 applications often delete files and create them again.

Resource security is described in Chapter 5.

Security Audit Journal

Several functions exist on the system to help you audit the effectiveness of security. In particular, the system provides the ability to log selected security-related events in a security audit journal. Several system values, user profile values, and object values control which events are logged.

Chapter 9 provides information about auditing security.

C2 Security

By using security level 50 and following the instructions in the *Guide to Enabling C2 Security*, you can bring your AS/400 system to a C2 level of security. C2 is a security standard defined by the U.S. government in the *Department of Defense Trusted System Evaluation Criteria* (DoD 5200.28.STD).

A C2 system enforces discretionary access control, user accountability, security auditing, and resource isolation.

Chapter 2. System Security Level

This chapter discusses the security level (QSECURITY) system value and the issues associated with it.

Overview

Purpose: Specify level of security to be enforced on the system.

How To: `WRKSYSVAL *SEC` (Work with System Values command) or Menu SETUP, option 1 (Change System Options)

Authority: *ALLOBJ and *SECADM

Journal Entry: SV

Notes: Change takes effect at next IPL. Before changing on a productive system, read appropriate section on migrating from one level to another.

The system offers five levels of security:

10 No system-enforced security

20 Sign-on security

30 Sign-on and resource security

40 Sign-on and resource security; integrity protection

50 Sign-on and resource security; enhanced integrity protection.

For ease of installation, your system is shipped at level 10, which provides no password or resource security. Any user (including remote users starting communications jobs) can use any resource.

After your system is installed, change the security level to at least level 20. Level 30 or higher is recommended. Use the Work with System Values (WRKSYSVAL) command to change the security level. The change takes effect the next time you perform an initial program load (IPL). Table 2-1 compares the levels of security on the system:

Table 2-1. Security Levels: Function Comparison

Function	**Level 10**	**Level 20**	**Level 30**	**Level 40**	**Level 50**
User name required to sign on.	Yes	Yes	Yes	Yes	Yes
Password required to sign on.	No	Yes	Yes	Yes	Yes
Password security active.	No	Yes	Yes	Yes	Yes
Menu and initial program security active.	No	Yes[1]	Yes[1]	Yes[1]	Yes[1]
Limit capabilities support active.	No	Yes	Yes	Yes	Yes
Resource security active.	No	No	Yes	Yes	Yes
Access to all objects.	Yes	Yes	No	No	No
User profile created automatically.	Yes	No	No	No	No
Security auditing capabilities available.	Yes	Yes	Yes	Yes	Yes
Programs that contain restricted instructions cannot be created or recompiled.	Yes	Yes	Yes	Yes	Yes
Programs that use unsupported interfaces fail at run time.	No	No	No	Yes	Yes
Enhanced hardware storage protection supported.	No	No	No	Yes	Yes
Library QTEMP is a temporary object.	No	No	No	No	Yes
*USRSPC, *USRIDX, and *USRQ objects can be created only in libraries specified in the QALWUSRDMN system value.	Yes	Yes	Yes	Yes	Yes
Pointers used in parameters are validated for user domain programs running in system state.	No	No	No	No	Yes
Message handling rules are enforced between system and user state programs.	No	No	No	No	Yes
A program's associated space cannot be directly modified.	No	No	No	Yes	Yes
Internal control blocks are protected.	No	No	No	Yes	Yes[2]

[1] When LMTCPB(*YES) is specified in the user profile.

[2] At level 50, more protection of internal control blocks is enforced than at level 40. See "Preventing Modification of Internal Control Blocks" on page 2-9.

The system security level determines what the default special authorities are for each user class. When you create a user profile, you can select special authorities based on the user class. Special authorities are also added and removed from user profiles when you change security levels.

These special authorities can be specified for a user:

*ALLOBJ All-object special authority gives a user authority to perform all operations on objects.

*SECADM Security administrator special authority allows a user to work with user profiles on the system.

*JOBCTL Job control special authority allows a user to control batch jobs and printing on the system.

*SPLCTL Spool control special authority allows unrestricted control of batch jobs and output queues on the system.

*SAVSYS Save system special authority allows a user to save and restore objects.

*SERVICE Service special authority allows a user to perform software service functions on the system.

*AUDIT Audit special authority allows a user to define the auditing characteristics of the system, objects, and system users.

Table 2-2 shows the default special authorities for each user class. The entries indicate that the authority is given at security levels 10 and 20 only, at all security levels, or not at all.

Table 2-2. Default Special Authorities for User Classes by Security Level

Special	User Classes				
Authority	*SECOFR	*SECADM	*PGMR	*SYSOPR	*USER
*ALLOBJ	All	10 or 20	10 or 20	10 or 20	10 or 20
*SECADM	All	All			
*JOBCTL	All	All	All	All	
*SPLCTL	All				
*SAVSYS	All	All	All	All	10 or 20
*SERVICE	All				
*AUDIT	All				

Note: The topics "User Class" on page 4-4 and "Special Authority" on page 4-7 provide more information about user classes and special authorities.

Recommendations: Security level 30 is recommended because the system does not automatically give users access to all resources. At lower security levels, all users are given *ALLOBJ special authority. Systems with higher security requirements should use security level 40 or 50.

Security level 50 is intended for systems with very high security requirements. If you run your system at security level 50, you may notice some performance impact because of the additional checking the system performs.

Even if you want to give all users access to all information, consider running your system at security level 30. You can use the public authority capability to give users access to information. Using security level 30 from the beginning gives you the flexibility of securing a few critical resources when you need to without having to test all your applications again.

Security Level 10

At security level 10, you have minimal security protection. When a new user signs on, the system creates a user profile with the profile name equal to the user ID specified on the sign-on display. If the same user signs on later with a different user ID, a new user profile is created. Appendix B shows the default values that are used when the system automatically creates a user profile.

The system performs authority checking at all levels of security. Because all user profiles created at security level 10 are given *ALLOBJ special authority, users pass every authority check and have access to all resources. If you want to test the effect of moving to a higher security level, you can remove *ALLOBJ special authority from user profiles and grant those profiles the authority to use specific resources. However, this does not give you any security protection. Anyone can sign on with a new user ID, and a new profile is created with *ALLOBJ special authority. You cannot prevent this at security level 10.

Security Level 20

Level 20 provides the following security functions, in addition to those provided at level 10:

- Both user ID and password are required to sign on.
- Only a security officer or someone with *SECADM special authority can create user profiles.
- The limit capabilities value specified in the user profile is enforced.

Changing to Level 20 from Level 10

When you change from level 10 to level 20, any user profiles that were automatically created at level 10 are preserved. The password for each user profile that was created at level 10 is the same as the user profile name. No changes are made to the special authorities in the user profiles.

Following is a recommended list of activities if you plan to change from level 10 to level 20 after your system has been in production:

- List all the user profiles on the system using the Display Authorized User (DSPAUTUSR) command.
- Either create new user profiles with standardized names or copy the existing profiles and give them new, standardized names.
- Set the password to expired in each existing profile, forcing each user to assign a new password.
- Set password composition system values to prevent users from assigning trivial passwords.
- Review the default values in Table B-1 in Appendix B for any changes you want to make to the profiles automatically created at security level 10.

Changing to Level 10 or 20 from a Higher Level

When you change from a higher security level to level 10 or 20, special authorities are added to and removed from user profiles to match the default special authorities for the user class. Refer to Table 2-2 to see how special authorities differ between level 10 or 20 and higher security levels.

Warning: When you change to level 10 or 20 from a higher security level, the system adds *ALLOBJ special authority to every user profile. This allows users to view, change, or delete any object on the system.

Security Level 30

Level 30 provides the following security functions, in addition to what is provided at level 20:

- Users must be specifically given authority to use resources on the system.
- Only user profiles created with the *SECOFR security class are given *ALLOBJ special authority automatically.

Changing to Level 30 from a Lower Level

When you change to security level 30 from a lower security level, the system changes all user profiles the next time you perform an IPL. Special authorities are added to and removed from user profiles to match the default special authorities for the user class. For example, *ALLOBJ special authority is removed from all user profiles except those with a user class of *SECOFR. Refer to Table 2-2 on page 2-2 to see how special authorities differ between level 10 or 20 and higher security levels.

If your system has been running applications at a lower security level, you should set up and test resource security before changing to security level 30. Following is a recommended list of activities:

- For each application, set the appropriate authorities for application objects.
- Test each application using either actual user profiles or special test user profiles:
  - Remove *ALLOBJ special authority from the user profiles used for testing.
  - Grant appropriate application authorities to the user profiles.
  - Run the application using the user profiles.
  - Check for authority failures either by looking for error messages or by using the security audit journal.
- When all applications run successfully with test profiles, grant the appropriate authorities for application objects to all production user profiles.
- If the QLMTSECOFR (limit security officer) system value is 1 (Yes), users with *ALLOBJ or *SERVICE special authority must be specifically authorized to devices at security level 30 or higher. Give these users *CHANGE authority to selected devices, give QSECOFR *CHANGE authority to the devices, or change the QLMTSECOFR system value to 0.
- Change the security level on your system and perform an initial program load (IPL).

If you want to change to level 30 without defining individual object authorities, make the public authority for application objects high enough to run the application. Run application tests to make sure no authority failures occur.

Note: See the topic “Defining How Information Can Be Accessed” on page 5-2 for more information about object authorities.

Security Level 40

Security level 40 prevents potential integrity or security risks from programs that could circumvent security in special cases. Security level 50 provides enhanced integrity protection for installations with strict security requirements. Table 2-3 on page 2-4 compares how security functions are supported at levels 30, 40, and 50. These functions are explained in more detail in the sections that follow.

Table 2-3. Comparison of Security Levels 30, 40, and 50

Scenario Description	**Level 30**	**Level 40**	**Level 50**
A program attempts to access objects using interfaces that are not supported.	AF journal entry [1]	AF journal entry [1]; operation fails.	AF journal entry [1]; operation fails.
A program attempts to use a restricted instruction.	AF journal entry [1]	AF journal entry [1]; operation fails.	AF journal entry [1]; operation fails.
The user submitting a job does not have *USE authority to the user profile specified in the job description.	AF journal entry [1]	AF journal entry [1]; job does not run.	AF journal entry [1]; job does not run.
A user attempts default sign-on without a user ID and a password.	AF journal entry [1]	AF journal entry [1]; sign-on is not successful.	AF journal entry [1]; sign-on is not successful.
A *USER state program attempts to write to system area of disk defined as read only or no access.	Attempt is successful.	AF journal entry; [1,2] operation fails. [2]	AF journal entry; [1,2] operation fails. [2]
An attempt is made to restore a program that does not have a validation value. [3]	No validation is performed. Program is restored with no ownership changes.	Program validation is performed.	Program validation is performed.
An attempt is made to restore a program that has a validation value.	Program validation is performed.	Program validation is performed.	Program validation is performed.
An attempt is made to modify a program's associated space.	Attempt is successful.	AF journal entry;[1,2] operation fails.[2]	AF journal entry;[1,2] operation fails.[2]
An attempt is made to modify a job's address space.	Attempt is successful.	AF journal entry;[1,2] operation fails.[2]	AF journal entry;[1,2] operation fails.[2]
An attempt is made to create a user domain object of type *USRSPC, *USRIDX, or *USRQ in a library not included in the QALWUSRDMN system value.	Operation fails.	Operation fails.	Operation fails.
A user state program sends an exception message to a system state program that is not immediately above it in the program stack.	Attempt is successful.	Attempt is successful.	Operation fails.
A parameter is passed to a user domain program running in the system state.	Attempt is successful.	Attempt is successful.	Parameter validation is performed.

[1] An authority failure (AF) type entry is written to the audit (QAUDJRN) journal, if the auditing function is active. See Chapter 9 for more information about the audit function.

[2] If the processor supports enhanced hardware storage protection.

[3] Programs created prior to Version 1 Release 3 do not have a validation value.

If you use the auditing function at lower security levels, the system logs journal entries for most of the actions shown in Table 2-3, except those detected by the enhanced hardware protection function. You receive warnings in the form of journal entries for potential integrity violations. At level 40 and higher, integrity violations cause the system to fail the attempted operation.

Preventing the Use of Unsupported Interfaces

At security level 40 and higher, the system prevents attempts to directly call system programs not documented as call-level interfaces. For example, directly calling the command processing program for the SIGNOFF command fails. See Appendix E, "Security APIs and Authority for Call Level Interfaces" for a list of call-level interfaces that are supported.

The system uses the domain attribute of an object and the state attribute of a program to enforce this protection:

Domain: Every object belongs to either the *SYSTEM domain or the *USER domain. *SYSTEM domain objects can be accessed only by *SYSTEM state programs.

You can display the domain of an object by using the Display Object Description (DSPOBJD) command and specifying DETAIL(*FULL). You can also use the Display Program (DSPPGM) command to display the domain of a program.

State: Every program is either a *SYSTEM state program or *USER state program. *USER state programs can directly access only *USER domain objects. Objects that are *SYSTEM domain can be accessed using the appropriate command or application programming interface (API). The *SYSTEM state is reserved for IBM-supplied programs.

You can display the state of a program using the Display Program (DSPPGM) command.

Table 2-4 shows the domain and state access rules:

Table 2-4. Domain and State Access

	Object Domain	
Program State	***USER**	***SYSTEM**
*USER	YES	NO [1]
*SYSTEM	YES	YES

[1] A domain or state violation causes the operation to fail at security level 40 and higher. At all security levels, an AF type entry is written to the audit journal if the auditing function is active.

Journal Entry: If the auditing function is active and the QAUDLVL system value includes *PGMFAIL, an authority failure (AF) entry, violation type D, is written to the QAUDJRN journal when an attempt is made to use an unsupported interface.

Preventing the Use of Restricted Instructions

At security level 40 and higher, the system prevents access to internal system structures through pointer capabilities of the C/400* programming language, Pascal, or Assembler.

Journal Entry: If the auditing function is active and the QAUDLVL system value includes *PGMFAIL, an authority failure (AF) entry, violation type B, is written to the QAUDJRN journal when an attempt is made to use a restricted instruction.

Protecting Job Descriptions

If a user profile name is used as the value for the *User* field in a job description, any jobs submitted with the job description can be run with attributes taken from that user profile. An unauthorized user might use a job description to violate security by submitting a job to run under the user profile specified in the job description.

At security level 40 and higher, the user submitting the job must have *USE authority to both the job description and the user profile specified in the job description, or the job fails. At security level 30, the job runs if the submitter has *USE authority to the job description.

Journal Entry: If the auditing function is active and the QAUDLVL system value includes *AUTFAIL, an AF entry, violation type J, is written to the QAUDJRN journal when a user submits a job and is not authorized to the user profile in a job description.

Signing On without Password

At security level 30 and below, signing on by pressing the Enter key without a user ID and password is possible with certain subsystem descriptions. At security level 40 and higher, the system stops any attempt to sign on without a user ID and password. See the topic "Security and Subsystem Descriptions" on page 6-3 for more information about security issues associated with subsystem descriptions.

Journal Entry: An AF entry, violation type S, is written to the QAUDJRN journal when a user attempts to sign on without entering a user ID and password and the subsystem description allows it. (The attempt fails at security level 40 and higher.)

Enhanced Hardware Storage Protection

Enhanced hardware storage protection allows blocks of system information located on disk to be defined as read-write, read only, or no access. At security level 40 and higher, the system controls how *USER state programs access these protected blocks. This support is not available at security levels less than 40.

Enhanced hardware storage protection is supported on all AS/400 models, *except* the following:

- All B models
- All C models
- D models: 9402 D04, 9402 D06, 9404 D10, and 9404 D20.

Journal Entry: If the auditing function is active and the QAUDLVL system value includes *PGMFAIL, an AF entry, violation type R, is written to the QAUDJRN journal when a program attempts to write to an area of disk protected by the enhanced hardware storage protection feature. This support is available only at security level 40 and higher.

Protecting a Program's Associated Space

At security level 40 and higher, a user state program cannot directly change the associated space of a program object.

Protecting a Job's Address Space

At security level 50, a user state program cannot obtain the address for another job on the system. Therefore, a user state program cannot directly manipulate objects associated with another job.

Validation of Programs Being Restored

Beginning with Version 1 Release 3 of the OS/400* licensed program, a program containing restricted instructions cannot be compiled or created on an AS/400 system. The system uses a technique called **program validation** to determine whether a program being restored to your system may contain restricted instructions, either because the program was created on an earlier release or because the object code has been changed.

When a program is created at Version 1 Release 3 or later, the AS/400 system calculates a validation value, which is stored with the program. When the program is restored, the validation value is calculated again and compared to the validation value that is stored with the program. If the validation values match, the program is restored.

If the validation values do not match, the actions taken by the system are determined by the security level and by the ALWOBJDIF parameter on the Restore Object (RSTOBJ) command. The system actions may be:

- Try to create the program again.
- Log an entry in the audit journal.
- Send a message to the job log.
- Change ownership of the restored program to QDFTOWN.
- Revoke authority to the restored program.

Figure 2-1 on page 2-7 shows the procedure used by the system to determine what action to take. On the figure, the process of re-creating the program is called **translation**, which means creating the object code again from the observable information stored with the object code. The program source is not required for translation. On the figure, Version 1 Release 3 is abbreviated as V1R3.

For programs created prior to Version 1 Release 3, you can use the Change Program (CHGPGM) command with the Force Create (FRCCRT) parameter to have the system create a validation value. This improves restore performance after migrating to security level 40 or higher.

If programs have the observable information deleted, the CHGPGM command cannot calculate a validation value. In this case, the validation value can only be created by using the CRTxxxPGM command.

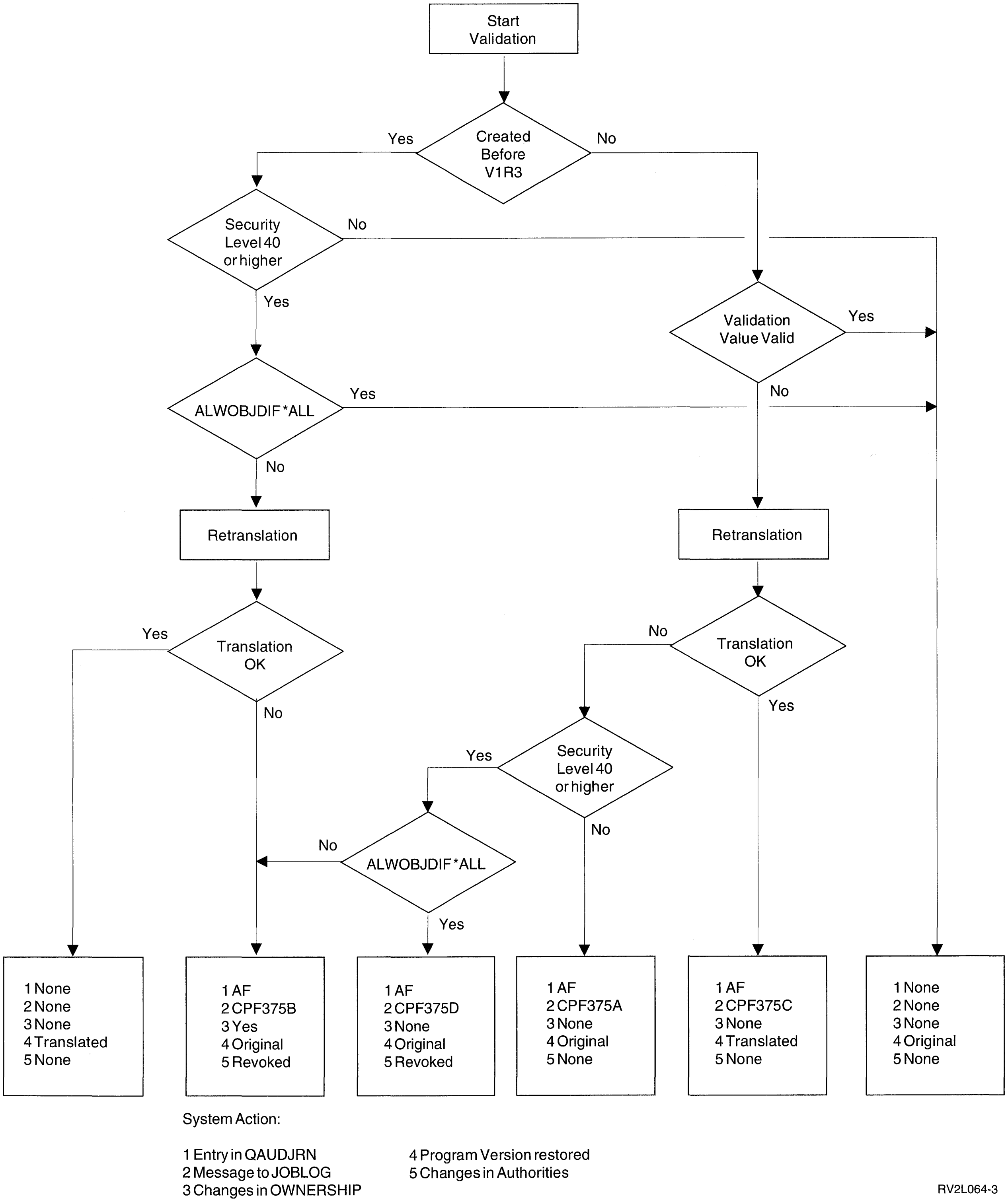


Figure 2-1. Validation Checking and System Action When Restoring a Program

Changing to Security Level 40

Make sure that all your applications run successfully at security level 30 before migrating to level 40. Security level 30 gives you the opportunity to test resource security for all your applications. Use the following procedure to migrate to security level 40:

1. Activate the security auditing function, if you have not already done so. The topic "Setting up Security Auditing" on page 9-10 gives complete instructions for setting up the auditing function.
2. Make sure the QAUDLVL system value includes *AUTFAIL and *PGMFAIL. *PGMFAIL logs journal entries for any access attempts that violate the integrity protection at security level 40.
3. Monitor the audit journal for *AUTFAIL and *PGMFAIL entries while running all your applications at security level 30. Pay particular attention to the following reason codes in AF type entries:

   B Restriction (blocked) instruction violation

   C Object validation failure

   D Unsupported interface (domain) violation

   J Job-description and user-profile authorization failure

   R Attempt to access protected area of disk (enhanced hardware storage protection)

   S Default sign-on attempt

   These codes indicate the presence of integrity exposures in your applications. At security level 40, these programs fail.
4. If you have any programs that were created prior to Version 1 Release 3, use the CHGPGM command with the FRCCRT parameter to create validation values for those programs. At security level 40, the system translates any program that is restored without a validation value. This can add considerable time to the restore process. See the topic "Validation of Programs Being Restored" on page 2-6 for more information about program validation.

   Note: Restore program libraries as part of your application test. Check the audit journal for validation failures.
5. Based on the entries in the audit journal, take steps to correct your applications and prevent program failures.
6. Change the QSECURITY system value to 40 and perform an IPL.

Disabling Security Level 40

After changing to security level 40, you may find you need to move back to level 30 temporarily. For example, you may need to test new applications for integrity errors. Or, you may discover you did not test well enough before changing to security level 40.

You can change from security level 40 to level 30 without jeopardizing your resource security. No changes are made to special authorities in user profiles when you move from level 40 to level 30. After you have tested your applications and resolved any errors in the audit journal, you can move back to level 40.

Warning: If you move from level 40 to level 20 or level 10, some special authorities are added to all user profiles. (See Table 2-2 on page 2-2.) This removes resource security protection.

Security Level 50

Security level 50 is designed to meet the requirements defined by the U.S. Department of Defense for C2 security. It provides enhanced integrity protection in addition to what is provided by security level 40. Running your system at security level 50 is required for C2 security. Other requirements for C2 security are described in the *Guide to Enabling C2 Security*.

These security functions are included for security level 50. They are described in the topics that follow:

- Restricting user domain object types (*USRSPC, *USRIDX, and *USRQ)
- Validating parameters
- Restricting message handling between user and system state programs
- Preventing modification of internal control blocks
- Making the QTEMP library a temporary object

Restricting User Domain Objects

Most object types on the system are created in system domain. When you run your system at security level 40 or 50, system domain objects can be accessed only by using the commands and APIs provided.

These object types can be either system or user domain:

- User space (*USRSPC)
- User index (*USRIDX)
- User queue (*USRQ)

Objects of type *USRSPC, *USRIDX, and *USRQ in user domain can be manipulated directly without using system-provided APIs and commands. This allows a user to access an object without creating an audit record.

Note: Objects of type *PGM, *SRVPGM and *SQLPKG can also be in the user domain. Their contents cannot be manipulated directly, and they are not affected by the restrictions.

At security level 50, a user must not be permitted to pass security-relevant information to another user without the ability to send an audit record. To enforce this:

- The QTEMP library for a job is a temporary object and cannot be accessed by another job. Therefore, if user domain objects are stored in the QTEMP library, they cannot be used to pass information to another user. The QTEMP library is a temporary object only if your security level is set to 50.

When the QTEMP library is a temporary object, the objects in QTEMP library may not be deleted by the system after an abnormal IPL or when a job ends abnormally. You may need to run the Reclaim Storage (RCLSTG) command more often at security level 50.

- To provide compatibility with existing applications that use user domain objects, you can specify additional libraries in the QALWUSRDMN system value. The QALWUSRDMN system value is enforced at all security levels. See "Allow User Domain Objects (QALWUSRDMN)" on page 3-1 for more information.

Validating Parameters

Interfaces to the operating system are system state programs in user domain. In other words, they are programs that can be called directly by a user. When parameters are passed between user state and system state programs, those parameters must be checked to prevent any unexpected values from jeopardizing the integrity of the operating system.

When you run your system at security level 50, the system specifically checks every parameter passed between a user state program and a system state program in the user domain. This is required for your system to separate the system and user domain and to meet the requirements of a C2 level of security. You may notice some performance impact because of this additional checking.

Restricting Message Handling

Messages sent between programs provide the potential for integrity exposures. The following applies to message handling at security level 50:

- Any user state program can send a message of any type to any other user state program.
- Any system state program can send a message of any type to any user or system state program.
- A user state program can send a non-exception message to any system state program.
- A user state program can send an exception type message (status, notify, or escape) to a system state program if one of the following is true:
  - The system state program is a request processor.
  - The system state program called a user state program.

    Note: The user state program sending the exception message does not have to be the program called by the system state program. For example, in this program stack, an exception message can be sent to Program A by Program B, C, or D:

Program A	System state
Program B	User state
Program C	User state
Program D	User state

- When a user state program receives a message from an external source (*EXT), any pointers in the message replacement text are removed.

Preventing Modification of Internal Control Blocks

At security level 40 and higher, some internal control blocks, such as the work control block and the system entry point table, cannot be modified by a user state program.

At security level 50, no internal control blocks can be modified. This includes the open data path (ODP), the spaces for CL commands and programs, and the S/36 environment job control block.

Changing to Security Level 50

Most of the additional security measures that are enforced at security level 50 do not cause audit journal entries at lower security levels. Therefore, an application cannot be tested for all possible integrity error conditions prior to changing to security level 50.

The actions that cause errors at security level 50 are uncommon in normal application software. Most software that runs successfully at security level 40 also runs at security level 50.

If you are currently running your system at security level 30, complete the steps described in "Changing to Security Level 40" on page 2-8 to prepare for changing to security level 50.

If you are currently running your system at security level 30 or 40, do the following to prepare for security level 50:

- Evaluate setting the QALWUSRDMN system value. Controlling user domain objects is important to system integrity. See "Restricting User Domain Objects" on page 2-8.
- Recompile any COBOL programs that assign the device in the SELECT clause to WORKSTATION. These programs must be recompiled using Version 2 Release 3 (V2R3) of the COBOL/400 licensed program to run successfully at security level 50.
- Recompile any S/36 environment COBOL programs using the V2R3 compiler.
- Recompile any RPG/400 programs that use display files. Use Version 2 Release 2 or later of the RPG/400 licensed program.

You can go directly from security level 30 to security level 50. Running at security level 40 as an intermediate step does not provide significant benefits for testing.

If you are currently running at security level 40, you can change to security level 50 without extra testing. Security level 50 cannot be tested in advance. The additional integrity protection that is enforced at security level 50 does not produce error messages or journal entries at lower security levels.

Disabling Security Level 50

After changing to security level 50, you may find you need to move back to security level 30 or 40 temporarily. For example, you may need to test new applications for integrity errors. Or, you may discover integrity problems that did not appear at lower security levels.

You can change from security level 50 to level 30 or 40 without jeopardizing your resource security. No changes are made to special authorities in user profiles when you move from level 50 to level 30 or 40. After you have tested your applications and resolved any errors in the audit journal, you can move back to level 50.

Warning: If you move from level 50 to level 20 or level 10, some special authorities are added to all user profiles. This removes resource security protection. (See Table 2-2 on page 2-2.)

Chapter 3. Security System Values

This chapter describes the system values that control security on your system. The system values are broken into four groups:

- General security system values
- Other system values related to security
- System values that control passwords
- System values that control auditing

General Security System Values

Overview

Purpose:	Specify system values that control security on the system.
How To:	`WRKSYSVAL *SEC` (Work with System Values command)
Authority:	*ALLOBJ and *SECADM
Journal Entry:	SV
Notes:	Changes take effect immediately. IPL is required only when changing the security level (QSECURITY system value).

Following are the general system values that control security on your system:

QALWUSRDMN	Allow user domain objects in the libraries
QCRTAUT	Create default public authority
QDSPSGNINF	Display sign-on information
QINACTITV [1]	Inactive job time-out interval
QINACTMSGQ [1]	Inactive job message queue
QLMTDEVSSN [1]	Limit device sessions
QLMTSECOFR [1]	Limit security officer
QMAXSIGN [1]	Maximum sign-on attempts
QMAXSGNACN [1]	Action when maximum sign-on attempts exceeded
QRMTSIGN	Remote sign-on requests
QSECURITY [1]	Security level

Descriptions of these system values follow. The possible choices are shown. The choices that are underlined are the system-supplied defaults. For most system values, a recommended choice is listed.

Allow User Domain Objects (QALWUSRDMN)

The QALWUSRDMN system value specifies which libraries are allowed to contain user domain objects of type *USRSPC, *USRIDX, and *USRQ. The restriction does not apply to user domain objects of type *PGM, *SRVPGM, and *SQLPKG. Systems with high security requirements require the restriction of user spaces, indexes, and queues. The system cannot audit the movement of information to and from user domain objects.

Possible Values for the QALWUSRDMN System Value:

<u>*ALL</u>	User domain objects of type *USRSPC, *USRIDX, and *USRQ are allowed in all libraries on the system.
library-name	The names of up to 50 libraries that can contain user domain objects of type *USRSPC, *USRIDX, and *USRQ. If individual libraries are listed, the library QTEMP *must* be included in the list.

Recommended Value: If your system has a high security requirement, you should allow user domain objects only in the QTEMP library. At security level 50, the QTEMP library is a temporary object and cannot be used to pass confidential data between users.

If your system has application software that relies upon objects of type *USRSPC, *USRIDX, or *USRQ, include the libraries used by the application software on the list of libraries for the QALWUSRDMN system value.

Note: If you run the Reclaim Storage (RCLSTG) command, user domain objects may need to be moved in and out of the QRCL (reclaim storage) library. To run the RCLSTG command successfully, you may need to add the QRCL library to the QALWUSRDMN system value. To protect system security, set the public authority to the QRCL library to *EXCLUDE. Remove the QRCL library from the QALWUSRDMN system value when you have finished running the RCLSTG command.

Authority for New Objects (QCRTAUT)

The QCRTAUT system value is used to determine the public authority for a newly created object if the following conditions are met:

- The create authority (CRTAUT) for the library of the new object is set to *SYSVAL.
- The new object is created with public authority (AUT) of *LIBCRTAUT.

[1] These system values are also discussed in the *Basic Security Guide*.

Possible Values for the QCRTAUT System Value:

<u>*CHANGE</u>	The public can change newly created objects.
***USE**	The public may view, but not change, newly created objects.
***ALL**	The public may perform any function on new objects.
***EXCLUDE**	The public is not allowed to use new objects.

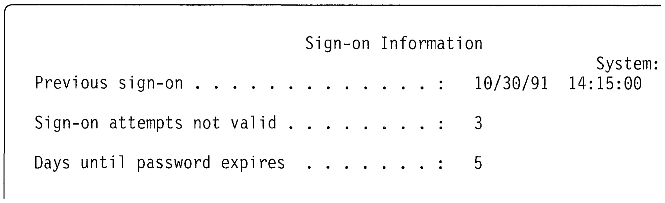
Recommended Value: *CHANGE

Warning: Several IBM-supplied libraries, including QSYS, have a CRTAUT value of *SYSVAL. If you change QCRTAUT to something other than *CHANGE, you may encounter problems. For example, the public authority for any new devices is controlled by QCRTAUT. If QCRTAUT is set to *USE or *EXCLUDE, public authority is not sufficient to allow signing on at new devices.

Display Sign-On Information (QDSPSGNINF)

The QDSPSGNINF system value determines whether the Sign-on Information display is shown after signing on. The Sign-on Information display shows:

- Date of last sign-on
- Any sign-on attempts that were not valid
- The number of days until the password expires (if the password is due to expire in 7 days or less)



```
                      Sign-on Information
                                                       System:
Previous sign-on . . . . . . . . . . . . . :   10/30/91  14:15:00

Sign-on attempts not valid . . . . . . . . :   3

Days until password expires  . . . . . . . :   5
```

Possible Values for the QDSPSGNINF System Value:

<u>0</u>	Display is not shown.
1	Display is shown.

Recommended Value: 1 (Display is shown) is recommended so users can monitor attempted use of their profiles and know when a new password is needed.

Note: Display sign-on information can also be specified in individual user profiles.

Inactive Job Time-Out Interval (QINACTITV)

The QINACTITV system value specifies in minutes how long the system allows a job to be inactive before taking action. A workstation is considered inactive if it is waiting at a menu or display, or if it is waiting for message input with no user interaction. Some examples of user interaction are:

- Using the Enter key
- Using the paging function
- Using function keys
- Using the Help key

Local jobs that are currently signed on to a remote system are excluded. PC Support/400* jobs are included.

Following are examples of how the system determines which jobs are inactive:

- A user runs a long job, such as a compile, interactively. After the user presses the Enter key to start the compile the workstation becomes inactive until the compile completes.
- A user uses the system request function to start a second interactive job. A system interaction, such as the Enter key, on either job causes both jobs to be marked as active.
- A PC Support/400 job may appear inactive to the system if the user is performing PC functions such as editing a document without interacting with the AS/400 system.

The QINACTMSGQ system value determines what action the system takes when an inactive job exceeds the specified interval.

When the system is started, it checks for inactive jobs at the interval specified by the QINACTITV system value. For example, if the system is started at 9:46 in the morning and the QINACTITV system value is 30 minutes, it checks for inactive jobs at 10:16, 10:46, 11:16, and so on. If it discovers a job that has been inactive for 30 minutes or more, it takes the action specified by the QINACTMSGQ system value. In this example, if a job becomes inactive at 10:17, it will not be acted upon until 11:16. At the 10:46 check, it has been inactive for only 29 minutes.

The QINACTITV and QINACTMSGQ system values provide security by preventing users from leaving inactive workstations signed on. An inactive workstation might allow an unauthorized person access to the system.

Possible Values for the QINACTITV System Value:

<u>*NONE:</u>	The system does not check for inactive jobs.
interval-in-minutes	Specify a value of 5 through 300. When a job has been inactive for that number of minutes, the system takes the action specified in QINACTMSGQ.

Recommended Value: 30 to 60 minutes.

Inactive Job Time-Out Message Queue (QINACTMSGQ)

The QINACTMSGQ system value specifies what action the system takes when the inactive job time-out interval for a job has been reached.

Possible Values for QINACTMSGQ System Value:

***ENDJOB** Inactive jobs are ended. If the inactive job is a group job, all jobs associated with the group are also ended. If the job is part of a secondary job, both jobs are ended. The action taken by *ENDJOB is equal to running the command `ENDJOB JOB(name) OPTION (*IMMED) ADLINTJOBS(*ALL)` against the inactive job.

***DSCJOB** The inactive job is disconnected, as are any secondary or group jobs associated with it. The disconnected job time-out interval (QDSCJOBITV) system value controls whether the system eventually ends disconnected jobs. See "Disconnected Job Time-Out Interval (QDSCJOBITV)" on page 3-5 for more information.

Warning: The system cannot disconnect some jobs, such as PC Organizer and PC text-assist function (PCTA). If the system cannot disconnect an inactive job, it ends the job instead.

message-queue-name Message CPI1126 is sent to the specified message queue when the inactive job time-out interval is reached. This message states: `Job &3/&2/&1 has not been active.`

The message queue must exist before it can be specified for the QINACTMSGQ system value. This message queue is automatically cleared during an IPL.

Recommended Value: *DSCJOB unless your users run PC Support/400 jobs. Using *DSCJOB when some PC Support/400 jobs are running is the equivalent of ending the jobs. It can cause significant loss of information. Use the *message-queue* option if you have the PC Support/400 licensed program. The *CL Programmer's Guide* shows an example of writing a program to handle messages.

Using a Message Queue: A user or a program can monitor the message queue and take action as needed, such as ending the job or sending a warning message to the user. Using a message queue allows you to make decisions about particular devices and user profiles, rather than treating all inactive devices in the same way. This method is recommended when you use the PC Support/400 licensed program.

If a workstation with two secondary jobs is inactive, two messages are sent to the message queue (one for each secondary job). A user or program can use the End Job (ENDJOB) command to end one or both secondary jobs. If an inactive job has one or more group jobs, a single message is sent to the message queue. Messages continue to be sent to the message queue for each interval that the job is inactive.

Limit Device Sessions (QLMTDEVSSN)

The QLMTDEVSSN system value specifies whether a user is allowed to be signed on to more than one device at a time. This value does not restrict the System Request menu or a second sign-on from the same device. If a user has a disconnected job, the user is allowed to sign on to the system with a new device session.

Possible Values for the QLMTDEVSSN System Value:

0 The system allows an unlimited number of sign-on sessions.

1 Users are limited to one device session.

Recommended Value: 1 (Yes) because limiting users to a single device reduces the likelihood of sharing passwords and leaving devices unattended.

Note: Limiting device sessions can also be specified in individual user profiles.

Limit Security Officer (QLMTSECOFR)

The QLMTSECOFR system value controls whether a user with all-object (*ALLOBJ) or service (*SERVICE) special authority can sign on to any workstation. Limiting powerful user profiles to certain well-controlled workstations provides security protection.

The QLMTSECOFR system value is only enforced at security level 30 and higher. "Security and Workstations" on page 6-2 provides more information about the authority required to sign on at a workstation.

You can always sign on at the system console with the QSECOFR profile, no matter how the QLMTSECOFR value is set.

Possible Values for the QLMTSECOFR System Value:

1 A user with *ALLOBJ or *SERVICE special authority can sign on at a display station only if that user is specifically authorized to the display station or if user profile QSECOFR is authorized to the display station.

0 Users with *ALLOBJ or *SERVICE special authority can sign on at any display station for which they have *CHANGE authority. They can receive *CHANGE authority through private or public authority.

Recommended Value: 1 (Yes).

Maximum Sign-On Attempts (QMAXSIGN)

The QMAXSIGN system value controls the number of consecutive sign-on attempts that are not correct by local and remote users. Incorrect sign-on attempts can be caused by a user ID that is not correct, a password that is not correct, or inadequate authority to use the workstation.

When the maximum number of sign-on attempts is reached, the QMAXSGNACN system value is used to determine the action to be taken. A message is sent to the QSYSOPR message queue (and QSYSMSG message queue if it exists in library QSYS) to notify the security officer of a possible intrusion.

If you create the QSYSMSG message queue in the QSYS library, messages about critical system events are sent to that message queue as well as to QSYSOPR. The QSYSMSG message queue can be monitored separately by a program or a system operator. This provides additional protection of your system resources. Critical system messages in QSYSOPR are sometimes missed because of the volume of messages sent to that message queue.

Possible Values for the QMAXSIGN System Value:

15 A user can try to sign on a maximum of 15 times.

***NOMAX** The system allows an unlimited number of incorrect sign-on attempts. This gives a potential intruder unlimited opportunities to guess a valid user ID and password combination.

limit Specify a value from 1 through 25. The recommended number of sign-on attempts is three. Usually three attempts are enough to correct typing errors but low enough to help prevent unauthorized access.

Recommended Value: 3.

Action When Sign-On Attempts Reached (QMAXSGNACN)

The QMAXSGNACN system value determines what the system does when the maximum number of sign-on attempts is reached at a workstation.

Possible Values for the QMAXSGNACN System Value:

3	Disable both the user profile and device.
1	Disable the device only.
2	Disable the user profile only.

The device is disabled only if the sign-on attempts that are not valid are consecutive on the same device. One valid sign-on resets the count of incorrect sign-on attempts for the device.

The user profile is disabled when the number of incorrect sign-on attempts for the user reaches the value in the QMAXSIGN system value, regardless of whether the incorrect sign-on attempts were from the same or different devices. One valid sign-on resets the count of incorrect sign-on attempts in the user profile.

If you create the QSYSMSG message queue in QSYS, the message sent (CPF1397) contains the user and device name. Therefore, it is possible to control the disabling of the device based on the device being used.

"Maximum Sign-On Attempts (QMAXSIGN)" on page 3-3 provides more information about the QSYSMSG message queue.

Recommended Value: 3.

Remote Sign-On Control (QRMTSIGN)

The QRMTSIGN system value specifies how the system handles remote sign-on requests. Examples of remote sign-on are display station pass-through from another system and the workstation function of the PC Support/400 licensed program.

Possible Values for the QRMTSIGN System Value:

***FRCSIGNON** Remote sign-on requests must go through the normal sign-on process.

***SAMEPRF** When the source and target user profile names are the same, the sign-on display may be bypassed if automatic sign-on is requested. Password verification occurs before the target pass-through program is used. If a password that is no valid is sent on an automatic sign-on attempt, the pass-through session always ends and an error message is sent to the user. However, if the profile names are different, *SAMEPRF indicates that the session ends with a security failure even i the user entered a valid password for the remote user profile.

The sign-on display appears for pass-through attempts not requesting automatic sign-on.

***VERIFY** The *VERIFY value allows you to bypass the sign-on display of the target system if valid security information is sent with the automatic sign-on request. If the password is not valid for the specified target user profile, the pass-through session ends with a security failure.

If the target system has a QSECURITY value of 10, any automatic sign-on request is allowed.

The sign-on display appears for pass-through attempts not requesting automatic sign-on.

***REJECT** No remote sign-on is permitted.

program-name library-name The program specified runs at the start and end of every pass-through session.

Recommended Value: *REJECT if you do not want to allow any pass-through or PC Support/400 access. If you do allow pass-through or PC Support/400 access, use *FRCSIGNON or *SAMEPRF.

The *Remote Work Station Guide* contains detailed information about the QRMTSIGN system value. It also contains the requirements for a remote sign-on program and an example.

Security-Related System Values

Overview

Purpose:	Specify system values that relate to security on the system.
How To:	`WRKSYSVAL` (Work with System Values command)
Authority:	*ALLOBJ and *SECADM
Journal Entry:	SV
Notes:	Changes take effect immediately. IPL is not required.

Following are descriptions of additional system values that relate to security on your system. These system values are not included in the *SEC group on the Work with System Values display.

QAUTOVRT	Automatic configuration of virtual devices
QDSCJOBITV [2]	Disconnected job time-out interval

Descriptions of these system values follow. For each value, the possible choices are shown. The choices that are underlined are the system-supplied defaults.

Automatic Configuration of Virtual Devices (QAUTOVRT)

The QAUTOVRT system value specifies whether pass-through virtual devices and TELNET full screen virtual devices (as opposed to the workstation function virtual device) are automatically configured.

A **virtual device** is a device description that does not have hardware associated with it. It is used to form a connection between a user and a physical workstation attached to a remote system.

Allowing the system to automatically configure virtual devices makes it easier for users to break into your system using pass-through. Without automatic configuration, a user attempting to break in has a limited number of attempts at each virtual device, the limit being defined by the security officer using the QMAXSIGN system value. With automatic configuration active, the actual limit is higher because the system sign-on limit is multiplied by the number of virtual devices that can be created by the automatic configuration support defined by the QAUTOVRT system value.

Possible Values for the QAUTOVRT System Value:

0	No virtual devices are created automatically.
number-of-virtual-devices	Specify a value 1 through 9999. If fewer than the specified number of devices are attached to a virtual controller and no device is available when a user attempts pass-through or full screen TELNET, the system configures a new device.

Recommended Value: 0.

The *Remote Work Station Guide*, has more information about using display station pass-through. The *TCP/IP Guide* has more information about using TELNET.

Disconnected Job Time-Out Interval (QDSCJOBITV)

The QDSCJOBITV system value determines if and when the system ends a disconnected job. The interval is specified in minutes.

If you set the QINACTMSGQ system value to disconnect inactive jobs (*DSCJOB), you should set the QDSCJOBITV to end the disconnected jobs eventually. A disconnected job uses up system resources, as well as retaining any locks on objects.

Possible Values for the QDSCJOBITV System Value:

240	The system ends a disconnected job after 240 minutes.
***NONE**	The system does not automatically end a disconnected job.
time-in-minutes	Specify a value between 5 and 1440.

Recommended Value: 120

System Values That Apply to Passwords

Overview

Purpose:	Specify system values to set requirements for the passwords users assign.
How To:	`WRKSYSVAL *SEC` (Work with System Values command)
Authority:	*ALLOBJ and *SECADM
Journal Entry:	SV
Notes:	Changes take effect immediately. IPL is not required.

Following are the system values that control passwords. These system values require users to change passwords regularly and help prevent users from assigning trivial, easily

[2] This system value is also discussed in the *Basic Security Guide*.

guessed passwords. They can also make sure passwords meet the requirements of your communications network:

QPWDEXPITV [3]	Expiration interval
QPWDMINLEN [3]	Minimum length
QPWDMAXLEN [3]	Maximum length
QPWDRQDDIF [3]	Required difference
QPWDLMTCHR	Restricted characters
QPWDLMTAJC	Restrict adjacent characters
QPWDLMTREP	Restrict repeating characters
QPWDPOSDIF	Character position difference
QPWDRQDDGT	Require numeric character
QPWDVLDPGM	Password validation program

The password-composition system values are enforced only when the password is changed using the CHGPWD command, the ASSIST menu option to change a password, or the QSYCHGPW application programming interface (API). They are not enforced when the password is set using the CRTUSRPRF or CHGUSRPRF command.

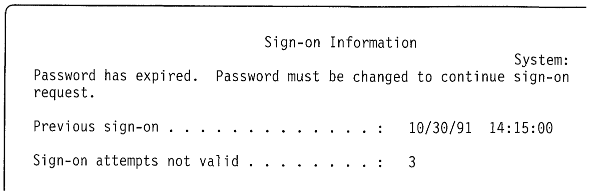
The default values supplied for the password-composition system values do not restrict the passwords a user can assign. Unless you change a particular system value, that password composition rule is not checked when the password is validated.

If you set any of the password-control system values, the system prevents a user from setting the password equal to the user profile name using the CHGPWD command, the ASSIST menu, or the QSYCHGPW API.

If a password is forgotten, the security officer can use the Change User Profile (CHGUSRPRF) command to set the password equal to the profile name or to any other value. The *Set password to expired* field in the user profile can be used to require that a password be changed the next time the user signs on.

Password Expiration Interval (QPWDEXPITV)

The QPWDEXPITV system value controls the number of days allowed before a password must be changed. If a user attempts to sign on after the password has expired, the system shows a display requiring that the password be changed before the user is allowed to sign on.



```
                          Sign-on Information
                                                               System:
Password has expired.  Password must be changed to continue sign-on
request.

Previous sign-on . . . . . . . . . . . . . :   10/30/91  14:15:00

Sign-on attempts not valid . . . . . . . . :   3
```

Possible Values for the QPWDEXPITV System Value:

*NOMAX	Users are not required to change their passwords.
limit-in-days	Specify a value from 1 through 366.

Recommended Value: 30 to 90.

Note: A password expiration interval can also be specified in individual user profiles.

Minimum Length of Passwords (QPWDMINLEN)

The QPWDMINLEN system value controls the minimum number of characters in a password.

Possible Values for the QPWDMINLEN System Value:

1	A minimum of one character is required for passwords.
minimum-number-of-characters	Specify a value of 1 through 10.

Recommended Value: 5, to prevent users from assigning passwords that are easily guessed, such as initials or a single character.

Maximum Length of Passwords (QPWDMAXLEN)

The QPWDMAXLEN system value controls the maximum number of characters in a password. This provides additional security by preventing users from specifying passwords that are too long and have to be recorded somewhere because they cannot be easily remembered.

Some communications networks require a password that is 8 characters or less. Use this system value to ensure that passwords meet the requirements of your network.

Possible Values for the QPWDMAXLEN System Value:

10	A maximum of ten characters for a password are allowed.
maximum-number-of-characters	Specify a value of 1 through 10

[3] These system values are also discussed in the *Basic Security Guide*.

Recommended Value: 8.

Required Difference in Passwords (QPWDRQDDIF)

The QPWDRQDDIF system value controls whether the password must be different than the 32 previous passwords. This value provides additional security by preventing users from specifying passwords used previously. It also prevents a user whose password has expired from changing it and then immediately changing it back to the old password.

Possible Values for the QPWDRQDDIF System Value:

0	Different passwords not required. A password can be the same as one of the previous 32 passwords.
1	A password cannot be the same as any of the previous 32 passwords.

Recommended Value: 1 (Yes).

Restricted Characters for Passwords (QPWDLMTCHR)

The QPWDLMTCHR system value limits the use of certain characters in a password. This value provides additional security by preventing users from using specific characters, such as vowels, in a password. Restricting vowels prevents users from forming actual words for their passwords.

Possible Values for the QPWDLMTCHR System Value:

*NONE	There are no restricted characters for passwords.
restricted-characters	Specify up to 10 restricted characters. The valid characters are A through Z, 0 through 9, and special characters pound (#), dollar ($), at (@), and underscore (_).

Recommended Value: A, E, I, O, and U. You may also want to prevent special characters (#, $, and @) for compatibility with other systems.

Restriction of Consecutive Digits for Passwords (QPWDLMTAJC)

The QPWDLMTAJC system value limits the use of numeric characters next to each other (adjacent) in a password. This value provides additional security by preventing users from using birthdays, telephone numbers, or a sequence of numbers as passwords.

Possible Values for the QPWDLMTAJC System Value:

0	Numeric characters are allowed next to each other in passwords.
1	Numeric characters are not allowed next to each other in passwords.

Restriction of Repeated Characters for Passwords (QPWDLMTREP)

The QPWDLMTREP system value limits the use of repeating characters in a password. This value provides additional security by preventing users from specifying the same character more than once in a password.

Possible Values for the QPWDLMTREP System Value:

0	The same characters can be used more than once in a password.
1	The same character cannot be used more than once in a password.

Character Position Difference for Passwords (QPWDPOSDIF)

The QPWDPOSDIF system value controls each position in a new password. This provides additional security by preventing users from using the same character (alphabetic or numeric) in a position corresponding to the same position in the previous password.

Possible Values for the QPWDPOSDIF System Value:

0	The same characters can be used in a position corresponding to the same position in the previous password.
1	The same character cannot be used in a position corresponding to the same position in the previous password.

Requirement for Numeric Character in Passwords (QPWDRQDDGT)

The QPWDRQDDGT system value controls whether a numeric character is required in a new password. This value provides additional security by preventing users from using all alphabetic characters.

Possible Values for the QPWDRQDDGT System Value:

0	Numeric characters are not required in new passwords.
1	One or more numeric characters are required in new passwords

Password Approval Program (QPWDVLDPGM)

If a program name is specified in the QPWDVLDPGM system value, the system runs that program after the new password has passed any validation tests you specify in the password-control system values. You can use the program to do additional checking of user-assigned passwords before they are accepted by the system.

The topic “Using a Password Approval Program” on page 3-8 discusses the requirements of the password approval program and shows an example.

Possible Values for the QPWDVLDPGM System Value:

***NONE** No user-written program is used.

program-name Specify the name of the user-written validation program, from 1 through 10 characters.

library-name Specify the name of the library where the user-written program is located. If the library name is not specified, the library list (*LIBL) of the user changing the system value is used to search for the program. QSYS is the recommended library.

Using a Password Approval Program: If a program name is specified in the QPWDVLDPGM system value, that program is called by the Change Password (CHGPWD) command. It is called only if the new password entered by the user has passed all the other tests you specified in the password-control system values.

In case it is necessary to recover your system from a disk failure, place the password approval program in library QSYS. This way the password approval program is loaded when you restore library QSYS.

The system passes the following parameters to the password approval program:

Table 3-1. Parameters for Password Approval Program

Position	Type	Length	Description
1	*CHAR	10	The new password entered by the user.
2	*CHAR	10	The user's old password.
3	*CHAR	1	Return code: 0 for valid password Not 0 for incorrect password

If your program determines that the new password is not valid, you can either send your own exception message (using the SNDPGMMSG command) or set the return code to a value other than 0 and let the system display an error message. Exception messages that are signaled by your program must be created with the DMPLST(*NONE) option of the Add Message Description (ADDMSGD) command.

The new password is accepted only if the user-written program ends with no escape message and a return code of 0. Because the return code is initially set for passwords that are not valid (not zero), the approval program must set the return code to 0 for the password to be changed.

Warning: The current and new password are passed to the validation program without encryption. The validation program could store passwords in a database file and compromise security on the system. Make sure the functions of the validation program are reviewed by the security officer and that changes to the program are strictly controlled.

The Display Password (DSPPWD) tool in the QUSRTOOL library uses a password validation program to save passwords and allow them to be displayed or listed by the security officer. The intent of the tool is to allow the security officer to monitor for trivial passwords. However, it represents a security exposure.

The following control language (CL) program is an example of a password approval program. This example checks to make sure the password is not changed more than once in the same day. Additional calculations can be added to the program to check other criteria for passwords:

```
/***************************************************/
/* NAME:     PWDVALID - Password Validation        */
/*                                                 */
/* FUNCTION: Limit password change to one per      */
/*           day unless the password is expired    */
/***************************************************/
   PGM (&NEW &OLD &RTNCD)
   DCL VAR(&NEW)        TYPE(*CHAR) LEN(10)
   DCL VAR(&OLD)        TYPE(*CHAR) LEN(10)
   DCL VAR(&RTNCD)      TYPE(*CHAR) LEN(1)
   DCL VAR(&JOBDATE)    TYPE(*CHAR) LEN(6)
   DCL VAR(&PWDCHGDAT) TYPE(*CHAR) LEN(6)
   DCL VAR(&PWDEXP)     TYPE(*CHAR) LEN(4)
/* Get the current date and convert to YMD format */
   RTVJOBA    DATE(&JOBDATE)
   CVTDAT     DATE(&JOBDATE) TOVAR(&JOBDATE) +
              TOFMT(*YMD)    TOSEP(*NONE)
/* Get date password last changed and whether      */
/* password is expired from user profile           */
   RTVUSRPRF  PWDCHGDAT(&PWDCHGDAT) PWDEXP(&PWDEXP)
/* Compare two dates                               */
/*    if equal and password not expired            */
/*    then send *ESCAPE message to prevent change */
/*    else set return code to allow change         */
   IF (&JOBDATE=&PWDCHGDAT *AND &PWDEXP='*NO ') +
      SNDPGMMSG  MSGID(CPF9898) MSGF(QCPFMSG) +
      MSGDTA('Password can be changed only +
              once per day) +
      MSGTYPE(*ESCAPE)
   ELSE  CHGVAR &RTNCD '0'
   ENDPGM
```

System Values That Control Auditing

Overview

Purpose: Specify system values to control security auditing on the system.

How To: `WRKSYSVAL *SEC` (Work with System Values command)

Authority: *AUDIT

Journal Entry: SV

Notes: Changes take effect immediately. IPL is not required.

These system values control auditing on the system:

QAUDCTL	Auditing control
QAUDENDACN	Auditing end action
QAUDFRCLVL	Auditing force level
QAUDLVL	Auditing level
QCRTOBJAUD	Create default auditing

Descriptions of these system values follow. The possible choices are shown. The choices that are underlined are the system-supplied defaults. For most system values, a recommended choice is listed.

Auditing Control (QAUDCTL)

The QAUDCTL system value determines whether auditing is performed. It functions like an on and off switch for the following:

- The QAUDLVL system value
- The auditing defined for objects using the Change Object Auditing (CHGOBJAUD) and Change DLO Auditing (CHGDLOAUD) commands
- The auditing defined for users using the Change User Audit (CHGUSRAUD) command

You can specify more than one value for the QAUDCTL system value, unless you specify *NONE.

Possible Values for the QAUDCTL System Value:

<u>*NONE</u> No auditing of user actions and no auditing of objects is performed.

***OBJAUD** Auditing is performed for objects that have been selected using the CHGOBJAUD and CHGDLOAUD commands.

***AUDLVL** Auditing is performed for any functions selected on the QAUDLVL system value and on the AUDLVL parameter of individual user profiles. The audit level for a user is specified using the Change User Audit (CHGUSRAUD) command.

Note: The QAUDCTL system value is available beginning with Version 2 Release 3 (V2R3) of the OS/400 licensed program. If auditing is active on your system (QAUDLVL is not *NONE) at an earlier release, the QAUDCTL system value is set to *AUDLVL when you move to V2R3.

See "Planning Security Auditing" on page 9-4 for a complete description of the process for controlling auditing on your system.

Auditing End Action (QAUDENDACN)

The QAUDENDACN system value determines what action the system takes if auditing is active and the system is unable to write entries to the audit journal.

Possible Values for the QAUDENDACN System Value:

<u>*NOTIFY</u> Message CPI2283 is sent to the QSYSOPR message queue and the QSYSMSG message queue (if it exists) every hour until auditing is successfully restarted. The system value QAUDCTL is set to *NONE to prevent the system from attempting to write additional audit journal entries. Processing on the system continues.

If an IPL is performed before auditing is restarted, message CPI2284 is sent to the QSYSOPR and QSYSMSG message queues during the IPL.

***PWRDWNSYS** If the system is unable to write an audit journal entry, the system powers down immediately. The system unit displays system reference code (SRC) B900 3D10. When the system is powered on again, it is in a restricted state. This means the controlling subsystem is in a restricted state, no other subsystems are active, and sign-on is allowed only at the console. The QAUDCTL system value is set to *NONE. The user who signs on the console to complete the IPL must have *ALLOBJ and *AUDIT special authority. See "Preventing Loss of Auditing Information" on page 9-10 for more information about restarting the system.

Recommended value: For most installations, *NOTIFY is the recommended value. If your security policy requires that no processing be performed on the system without auditing, then you must select *PWRDWNSYS.

Only very unusual circumstances cause the system to be unable to write audit journal entries. However, if this does happen and the QAUDENDACN system value is *PWRDWNSYS, your system ends abnormally. This could cause a lengthy initial program load (IPL) when your system is powered on again.

Auditing Force Level (QAUDFRCLVL)

The QAUDFRCLVL system value determines how often new audit journal entries are forced from memory to auxiliary storage. This system value controls the amount of auditing data that may be lost if the system ends abnormally.

Possible Values for the QAUDFRCLVL System Value:

<u>*SYS</u> The system determines when journal entries are written to auxiliary storage based on internal system performance.

Possible Values for the QAUDFRCLVL System Value:

number-of-records	Specify a number between 1 and 100 to determine how many audit entries can accumulate in memory before they are written to auxiliary storage. The smaller the number, the greater the impact on system performance.

Recommended value: *SYS provides the best auditing performance. However, if your installation requires that no audit entries be lost when your system ends abnormally, you must specify 1. Specifying 1 may impair performance.

Auditing Level (QAUDLVL)

The QAUDLVL system value determines which security-related events are logged to the security audit journal (QAUDJRN) for all system users. You can specify more than one value for the QAUDLVL system value, unless you specify *NONE.

For the QAUDLVL system value to take effect, the QAUDCTL system value must include *AUDLVL.

Possible Values for the QAUDLVL System Value:

***NONE**	No events controlled by the QAUDLVL system value are logged. Events are logged for individual users based on the AUDLVL values of user profiles.
***AUTFAIL**	Authority failure events are logged.
***CREATE**	Object create operations are logged.
***DELETE**	Object delete operations are logged.
***JOBDTA**	Actions that affect a job are logged.
***OBJMGT**	Object move and rename operations are logged.
***OFCSRV**	Changes to the system distribution directory and office mail actions are logged.
***PGMADP**	Obtaining authority from a program that adopts authority is logged.
***PGMFAIL**	System integrity violations are logged.
***PRTDTA**	Printing a spooled file and sending output directly to a printer are logged.
***SAVRST**	Restore operations are logged.
***SECURITY**	Security-related functions are logged.
***SERVICE**	Using service tools is logged.
***SPLFDTA**	Actions performed on spooled files are logged.
***SYSMGT**	Use of system management functions is logged.

See “Planning the Auditing of Actions” on page 9-4 for a complete description of the journal entry types and the possible values for QAUDLVL.

Auditing for New Objects (QCRTOBJAUD)

The QCRTOBJAUD system value is used to determine the auditing value for a new object, if the auditing default for the library of the new object is set to *SYSVAL. The QCRTOBJAUD system value is also the default object auditing value for new folderless documents.

For example, the CRTOBJAUD value for the CUSTLIB librar is *SYSVAL. The QCRTOBJAUD value is *CHANGE. If yo create a new object in the CUSTLIB library, its object auditing value is automatically set to *CHANGE. You can change the object auditing value using the CHGOBJAUD command.

Possible Values for the QCRTOBJAUD System Value:

***NONE**	No auditing is done for the object.
***USRPRF**	Auditing of the object is based on the value in the profile of the user accessing the object.
***CHANGE**	An audit record is written whenever the object i changed.
***ALL**	An audit record is written for any action that affects the contents of the object. An audit record is also written if an object’s contents change.

Recommended value: The value you select depends upon the auditing requirements of your installation. The section “Planning the Auditing of Object Access” on page 9-9 provides more information about methods for setting up object auditing on your system.

Chapter 4. User Profiles

This chapter describes user profiles: their purpose, their features, and how to design them. User profiles are a powerful and flexible tool. Designing them well can help you protect your system and customize it for your users.

Overview

Purpose:	To create and maintain user profiles and group profiles on the system.
How To:	Work with User Profiles (WRKUSRPRF) command
	Change User Audit (CHGUSRAUD) command
Authority:	*SECADM special authority
	*AUDIT special authority to change user auditing
Journal Entry:	CP
	AD for changes to user auditing
	ZC for changes to a user profile that are not relevant to security

Roles of the User Profile

The user profile has several roles on the system:

- It contains security-related information that controls how the user signs on the system, what the user is allowed to do after signing on, and how the user's actions are audited.
- It contains information that is designed to customize the system and adapt it to the user.
- It is a management and recovery tool for the operating system. The user profile contains information about the objects owned by the user and all the private authorities to objects.
- The user profile name identifies the user's jobs and printer output.

If the security level (QSECURITY) system value on your system is 10, the system automatically creates a user profile when someone signs on with a user ID that does not already exist on the system. Table B-1 in Appendix B shows the values assigned when the system creates a user profile.

If the QSECURITY system value on your system is 20 or higher, a user profile must exist before a user can sign on.

Group Profiles

A group profile is a special type of user profile. It serves two purposes on the system:

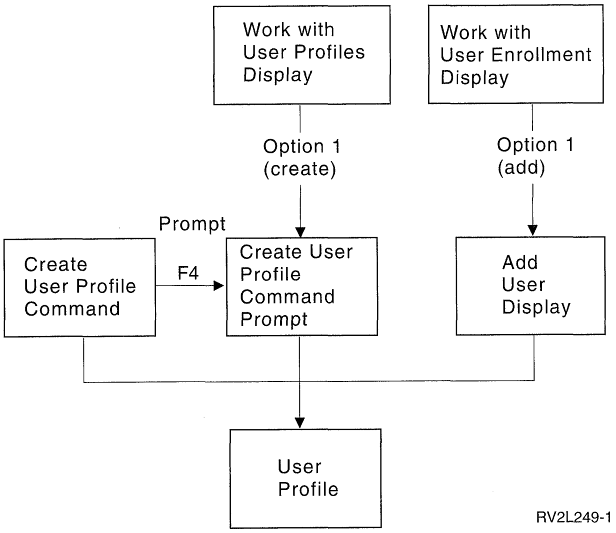
Security tool
A group profile provides a method for organizing authorities on your system and sharing them among users. You can define object authorities for group profiles rather than for each individual user profile.

Customizing tool
A group profile can be used as a pattern for creating individual user profiles. Most people who are part of the same group have the same customizing needs, such as the initial menu and the default printer. You can define these things in the group profile and then copy the group profile to create individual user profiles.

You create group profiles in the same way that you create individual profiles. The system recognizes a group profile when you add the first member to it. At that point, the system sets information in the profile indicating that it is a group profile. "Planning Group Profiles" on page 7-10 shows an example of setting up a group profile.

User-Profile Fields

User profiles can be created in several ways:



Work with User Profiles Display
Work with User Enrollment Display
Option 1 (create)
Option 1 (add)
Prompt
Create User Profile Command
F4
Create User Profile Command Prompt
Add User Display
User Profile
RV2L249-1

Figure 4-1. How User Profiles Are Created

Following are explanations of each field in the user profile. The fields are described in the order they appear on the Create User Profile command prompt.

Many system displays have different versions, called **assistance levels**, to meet the needs of different users:

- Basic assistance level, which contains less information and does not use technical terminology.
- Intermediate assistance level, which shows more information and uses technical terms.

- Advanced assistance level, which uses technical terms and shows the maximum amount of data by not always displaying function key and option information.

The sections that follow show what the user profile fields are called on both the basic assistance level and intermediate assistance level displays. This is the format used:

Field Title
The title of the section shows how the field name appears on the Create User Profile command prompt, which is shown when you create a user profile with intermediate assistance level or the Create User Profile (CRTUSRPRF) command.

Add User prompt:
This shows how the field name appears on the Add User display and other user-profile displays that use basic assistance level. The basic assistance level displays show a subset of the fields in the user profile. *Not shown* means the field does not appear on the basic assistance level display. When you use the Add User display to create a user profile, default values are used for all fields that are not shown.

CL parameter:
You use the CL parameter name for a field in a CL program or when you enter a user profile command without prompting.

Length:
If you use the Retrieve User Profile (RTVUSRPRF) command in a CL program, this is the length you should use to define the parameter associated with the field.

Authority:
If a field refers to a separate object, such as a library or a program, you are told the authority requirements for the object. To specify the object when you create or change a user profile, you need the authority listed. To sign on using the profile, the user needs the authority listed. For example, if you create user profile USERA with job description JOBD1, you must have *USE authority to JOBD1. USERA must have *USE authority to JOBD1 to successfully sign on with the profile.

In addition, each section describes the possible values for the field and a recommended value.

User Profile Name

Add User prompt:	User
CL parameter:	USRPRF
Length:	10

The user profile name identifies the user to the system. This user profile name is also known as the user ID. It is the name the user types in the *User* prompt on the Sign On display.

The user profile name can be a maximum of 10 characters. The characters can be:

- Any letter (A through Z)
- Any number (0 through 9)
- These special characters: pound (#), dollar ($), underscore (_), at (@).

Note: The Add User display allows only an eight-character user name.

The user profile name cannot begin with a number.

Note: It is possible to create a user profile so that when a user signs on, the user ID is only numerals. To create a profile like this, specify a Q as the first character, such as Q12345. A user can then sign on by entering 12345 or Q12345 for the *User* prompt on the Sign On display.

For more information about specifying names on the system, see the *CL Programmer's Guide*.

Recommendations for Naming User Profiles: Consider these things when deciding how to name user profiles:

- A user profile name can be up to 10 characters long. Both the OfficeVision/400 licensed program and some communications methods limit the user ID to eight characters. The Add User display also limits the user profile name to eight characters.

  Use eight characters or less if you plan to use the OfficeVision/400 licensed program or communications now or in the future.
- When you use the OfficeVision/400 licensed program, you send mail to a person's user ID. Use a naming scheme that makes user IDs easy to remember.
- The system does not distinguish between uppercase and lowercase letters in a user profile name. If you enter lowercase alphabetic characters at your workstation, the system translates them to uppercase characters.
- The displays and lists you use to manage user profiles show them in alphabetical order by user profile name.
- Avoid using special characters in user profile names. Special characters may cause problems with keyboard mapping for certain workstations or with national language versions of the OS/400 licensed program.

The *Planning For and Setting Up OfficeVision/400** manual provides more information about planning OfficeVision/400 users.

One technique for assigning user profile names (and OfficeVision/400 user IDs) is to use the first seven characters of the last name followed by the first character of the first name. For example:

User Name	User Profile Name
Anderson, George	ANDERSOG
Anderson, Roger	ANDERSOR
Harrisburg, Keith	HARRISBUK
Jones, Sharon	JONESS
Jones, Keith	JONESK

Recommendations for Naming Group Profiles: If you want to be able to easily identify group profiles on lists and displays, use a naming convention. Begin all group profile names with the same characters, such as GRP (for group) or DPT (for department).

Password

Add User prompt: Password

CL parameter: PASSWORD

Length: 10

The password is used to verify a user's authority to sign on the system. A user ID and a password must be specified to sign on when password security is active (QSECURITY system value is 20 or higher).

Passwords can be a maximum of 10 characters. The minimum and maximum length for passwords on your system are set by the QPWDMINLEN and QPWDMAXLEN system values. The rules for specifying passwords are the same as those used for user profile names. You can create an all-numeric password by specifying Q as the first character. If a user specifies Q12345 as the password on the Change Password display, the user can specify either 12345 or Q12345 as the password on the Sign On display.

One-way encryption is used to store the password on the system. No method is available to decode it. If a password is forgotten, the security officer can use the Change User Profile (CHGUSRPRF) command to assign a temporary password and set that password to expired, requiring the user to assign a new password at the next sign-on.

You can set system values to control the passwords that users assign. The password composition system values apply only when a user changes a password using the Change Password (CHGPWD) command, the Change password option from the ASSIST menu, or the QSYCHGPW API. If any password composition system values have been set, the user cannot set the password equal to the user profile name using the CHGPWD command, the ASSIST menu, or the QSYCHGPW API.

The password composition system values do not apply when the password is changed using the CHGUSRPRF command. This allows a user with *SECADM special authority to set a forgotten password to the user profile name or a trivial value and require the user to change it when signing on.

See the topic "System Values That Apply to Passwords" on page 3-5 for information about setting the password composition system values.

Possible Values for PASSWORD:

***USRPRF** The password for this user is the same as the user profile name.

***NONE** No password is assigned to this user profile. Sign-on is not allowed with this user profile if your system is at security level 20 or higher. You can submit a batch job using a user profile with password *NONE if you have proper authority to the user profile.

user-password An alphanumeric character string (10 characters or less).

Recommendations for Passwords:

- Set the password for a group profile to *NONE. This prevents anyone from signing on with the group profile, unless the system is at security level 10.
- When creating an individual user profile, set the password to an initial value and require a new password to be assigned when the user signs on (set password to expired *YES). The default password when creating a user profile is the same as the user profile name.
- If you use a trivial or default password when creating a new user profile, make sure the user intends to sign on immediately. If you expect a delay before the user signs on, set the status of the user profile to *DISABLED. Change the status to *ENABLED when the user is ready to sign on. This protects a new user profile from being used by someone who is not authorized.
- Use the password composition system values to prevent users from assigning trivial passwords.
- Some communications methods send passwords between systems and limit the password to eight characters. If your system communicates with other systems, use the QPWDMAXLEN system value to limit passwords to eight characters.

Set Password to Expired

Add User prompt: Not shown

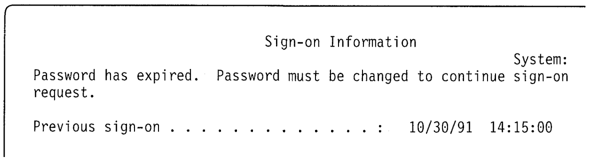
CL parameter: PWDEXP

Length: 4

The *Set password to expired* field allows a security administrator to indicate in the user profile that the user's password is expired and must be changed the next time the user signs on. This value is reset to *NO when the user changes the password using the CHGPWD or CHGUSRPRF command or as part of the next sign-on process.

This field can be used when a user cannot remember the password and a security administrator must assign a new one. Requiring the user to change the password assigned by the security administrator prevents the security administrator from knowing the new password and signing on as the user.

When a user's password has expired, the user receives a message at sign-on (see Figure 4-2). The user can either press the Enter key to assign a new password or press F3 (Exit) to cancel the sign-on attempt without assigning a new password. If the user chooses to change the password, the Change Password display is shown and password validation is run for the new password.



```
                       Sign-on Information
                                                           System:
Password has expired.  Password must be changed to continue sign-on
request.

Previous sign-on . . . . . . . . . . . . . :   10/30/91  14:15:00
```

Figure 4-2. Password Expiration Message

Possible Values for PWDEXP:

***NO:**	The password is not set to expired.
***YES:**	The password is set to expired.

Recommendations: Set the password to expired whenever you create a new user profile or assign a temporary password to a user.

Status

Add User prompt: Not shown

CL parameter: STATUS

Length: 10

The value of the *Status* field indicates if the profile is valid for sign-on. If the profile status is enabled, the profile is valid for sign-on. If the profile status is disabled, an authorized user has to enable the profile again to make it valid for sign-on.

You can use the CHGUSRPRF command to enable a profile that has been disabled. You must have *SECADM special authority and *OBJMGT and *USE authority to the profile to change its status. The topic "Enabling a User Profile" on page 4-21 shows an example of an adopted authority program to allow a system operator to enable a profile.

The system may disable a profile after a certain number of incorrect sign-on attempts with that profile, depending on the settings of the QMAXSIGN and QMAXSGNACN system values.

You can always sign on with the QSECOFR (security officer) profile at the console, even if the status of QSECOFR is *DISABLED. If the QSECOFR user profile becomes disabled, sign on as QSECOFR at the console and type `CHGUSRPRF QSECOFR STATUS(*ENABLED)`.

Possible Values for STATUS:

***ENABLED**	The profile is valid for sign-on.
***DISABLED**	The profile is not valid for sign-on until an authorized user enables it again.

Recommendations: Set the status to *DISABLED if you want to prevent sign-on with a user profile. For example, you can disable the profile of a user who will be away from the business for an extended period.

User Class

Add User prompt: Type of User

CL parameter: USRCLS

Length: 10

User class is used to control what menu options are shown to the user on OS/400 menus. This does not necessarily limit the use of commands. The *Limit capabilities* field controls whether the user can enter commands. User class may not affect what options are shown on menus provided by other licensed programs.

If no special authorities are specified when a user profile is created, the user class and the security level (QSECURITY) system value are used to determine the special authorities for the user.

Possible Values for USRCLS: Table 4-1 shows the possible user classes and what the default special authorities are for each user class. The entries indicate that the authority is given at security levels 10 and 20 only, at all security levels, or not at all.

The default value for user class is ***USER**.

Table 4-1. Default Special Authorities by User Class

Special	User Classes				
Authority	***SECOFR**	***SECADM**	***PGMR**	***SYSOPR**	***USER**
*ALLOBJ	All	10 or 20	10 or 20	10 or 20	10 or 20
*SECADM	All	All			
*JOBCTL	All	All	All	All	
*SPLCTL	All				
*SAVSYS	All	All	All	All	10 or 20
*SERVICE	All				
*AUDIT	All				

Recommendations: Most users do not need to perform system functions. Set the user class to *USER, unless a user specifically needs to use system functions.

Assistance Level

Add User prompt: Not shown

CL parameter: ASTLVL

Length: 10

For each user, the system keeps track of the last assistance level used for every system display that has more than one assistance level. That level is used the next time the user requests that display. During an active job, a user can change the assistance level for a display or group of related displays by pressing F21 (Select assistance level). The new assistance level for that display is stored with the user information.

Specifying the assistance level (ASTLVL) parameter on a command does not change the assistance level that is stored for the user for the associated display.

The *Assistance level* field in the user profile is used to specify the default assistance level for the user when the profile is created. If the assistance level in the user profile is changed using the CHGUSRPRF or the Change Profile (CHGPRF) command, the assistance levels stored for all displays for that user are reset to the new value.

For example, assume the user profile for USERA is created with the default assistance level (basic). Table 4-2 shows whether USERA sees the Work with User Profiles display or the Work with User Enrollment display when using different options. The table also shows whether the system changes the version for the display that is stored with USERA's profile.

Table 4-2. How Assistance Levels Are Stored and Changed

Action Taken	Version of Display Shown	Version of Display Stored
Use WRKUSRPRF command	Work with User Enrollment display	No change (basic assistance level)
From Work with User Enrollment display, press F21 and select intermediate assistance level.	Work with User Profiles display	Changed to intermediate assistance level
Use WRKUSRPRF command	Work with User Profiles display	No change (intermediate)
Select the work with user enrollment option from the SETUP menu.	Work with User Profiles display	No change (intermediate)
Type CHGUSRPRF USERA ASTLVL(*BASIC)		Changed to basic assistance level
Use WRKUSRPRF command	Work with User Enrollment display	No change (basic)
Type WRKUSRPRF ASTLVL(*INTERMED)	Work with User Profiles display	No change (basic)

Note: The *User option* field in the user profile also affects how system displays are shown. This field is described on page 4-16.

Possible Values for ASTLVL:

<u>*SYSVAL</u> The assistance level specified in the QASTLVL system value is used.

***BASIC** The Operational Assistant user interface is used.

***INTERMED** The system interface is used.

***ADVANCED** The expert system interface is used. To allow for more list entries, the option numbers and the function keys are not always displayed. If a command does not have an advanced (*ADVANCED) level, the intermediate (*INTERMED) level is used.

Current Library

Add User prompt:	Default library
CL parameter:	CURLIB
Length:	10
Authority	*USE

The current library is searched before the libraries in the user portion of the library list for any objects specified as *LIBL. If the user creates objects and specifies *CURLIB, the objects are put in the current library.

The current library is automatically added to the user's library list when the user signs on. It does not need to be included in the initial library list in the user's job description.

The user cannot change the current library if the *Limit capabilities* field in the user profile is *YES.

The topic "Security and Library Lists" on page 6-4 provides more information about using library lists and the current library.

Possible Values for CURLIB:

<u>*CRTDFT</u> This user has no current library. If objects are created using *CURLIB on a create command, the library QGPL is used as the default current library.

current-library-name The name of a library.

Recommendations: Use the *Current library* field to control where users are allowed to put new objects, such as Query programs. Use the *Limit capabilities* field to prevent users from changing the current library.

Initial Program

Add User prompt:	Sign on program
CL parameter:	INLPGM
Length:	10 (program name) 10 (library name)
Authority:	*USE for program *READ for library

You can specify the name of a program to call when a user signs on. This program runs before the initial menu, if any, is displayed. If the *Limit capabilities* field in the user's profile is *YES, the user cannot specify an initial program on the Sign On display.

The initial program is called only if the user's routing program is QCMD or QCL. See "Starting an Interactive Job" on page 6-1 for more information about the processing sequence when a user signs on.

Initial programs are used for two main purposes:

- To restrict a user to a specific set of functions.
- To perform some initial processing, such as opening files or establishing the library list, when the user first signs on.

Parameters cannot be passed to an initial program. If the initial program fails, the user is not able to sign on.

Possible Values for INLPGM:

*NONE	No program is called when the user signs on. If a menu name is specified on the initial menu (INLMNU) parameter, that menu is displayed.
program-name	The name of the program that is called when the user signs on.

Possible Values for INLPGM Library:

*LIBL	The library list is used to locate the program. If the job description for the user profile has an initial library list, that list is used. If the job description specifies *SYSVAL for the initial library list, the QUSRLIBL system value is used.
*CURLIB	The current library specified in the user profile is used to locate the program. If no current library is specified, QGPL is used.
library-name	The library where the program is located.

Initial Menu

Add User prompt:	First menu
CL parameter:	INLMNU
Length:	10 (menu name) 10 (library name)
Authority	*USE for menu *READ for library

You can specify the name of a menu to be shown when the user signs on. The initial menu is displayed after the user's initial program runs. The initial menu is called only if the user's routing program is QCMD or QCL.

If you want the user to run only the initial program, you can specify *SIGNOFF for the initial menu.

If the *Limit capabilities* field in the user's profile is *YES, the user cannot specify a different initial menu on the Sign On display. If a user is allowed to specify an initial menu on the Sign On display, the menu specified overrides the menu in the user profile.

Possible Values for MENU:

MAIN	The AS/400 system Main Menu is shown.
*SIGNOFF	The system signs off the user when the initial program completes. Use this to limit users to running a single program.
menu-name	The name of the menu that is called when the user signs on.

Possible Values for MENU Library:

*LIBL	The library list is used to locate the menu. If the initial program adds entries to the library list, those entries are included in the search, because the menu is called after the initial program has completed.
*CURLIB	The current library for the job is used to locate the menu. If no current library entry exists in the library list, QGPL is used.
library-name	The library where the menu is located.

Limit Capabilities

Add User prompt:	Restrict command line use
CL parameter:	LMTCPB
Length:	10

You can use the *Limit capabilities* field to limit the user's ability to enter commands and to override the initial program, initial menu, current library, and attention-key-handling program specified in the user profile. This field is an important security tool for preventing users from experimenting on the system.

A user with LMTCPB(*YES) can only run commands that are defined as allow limited user (ALWLMTUSR) *YES. These commands are shipped by IBM with ALWLMTUSR(*YES):

Sign off (SIGNOFF)
Send message (SNDMSG)
Display messages (DSPMSG)
Display job (DSPJOB)
Display job log (DSPJOBLOG)
Start PC Organizer (STRPCO)

The *Limit capabilities* field in the user profile and the ALWLMTUSR parameter on commands apply only to commands that are run from the command line or the Command Entry display. They do not restrict users from running commands in CL programs.

You can allow the limited capability user to run additional commands, or remove some of these commands from the list, by changing the ALWLMTUSR parameter for a command. Use the Change Command (CHGCMD) command. If you create your own commands, you can specify the ALWLMTUSR parameter on the Create Command (CRTCMD) command.

The Check Limit Capabilities (CHKLMTCPB) tool in the QUSRTOOL library provides a simple method of determining which users with user class *USER have the LMTCPB parameter specified as *NO. The tool also gives you the option to change all user profiles with user class *USER to LMTCPB(*YES).

Possible Values: Table 4-3 shows the possible values for *Limit capabilities* and what functions are allowed for each value.

Table 4-3. Functions Allowed for Limit Capabilities Values

Function	*YES	*PARTIAL	*NO
Change Initial Program	No	No	Yes
Change Initial Menu	No	Yes	Yes
Change Current Library	No	Yes	Yes
Change Attention Program	No	No	Yes
Enter Commands	A few[1]	Yes	Yes

[1] These commands are allowed: SIGNOFF, SNDMSG, DSPMSG, DSPJOB, DSPJOBLOG, and STRPCO. The user cannot use F9 to display a command line from any Operational Assistant menu or display.

Recommendations: Using an initial menu, restricting command line use, and providing access to the Operational Assistant* menu allow you to set up an environment for a user who does not need or want to access system functions. See the topic "Planning Menus" on page 7-4 for more information about this type of environment.

Text

Add User prompt:	User description
CL parameter:	TEXT
Length:	50

The text in the user profile is used to describe the user profile or what it is used for. For user profiles, the text should have identifying information, such as the user's name and department. For group profiles, the text should identify the group, such as what departments the group includes.

Possible Values for text:

*BLANK:	No text is specified.
description	Specify no more than 50 characters.

Recommendations: The *Text* field is truncated on many system displays. Put the most important identifying information at the beginning of the field.

Special Authority

Add User prompt:	Not shown
CL parameter:	SPCAUT
Length:	100 (10 characters per special authority)
Authority:	To give a special authority to a user profile, you must have that special authority.

Special authority is used to specify the type of actions a user can perform on system resources. A user can be given one or more special authorities.

Possible Values for SPCAUT:

*USRCLS	Special authorities are granted to this user based on the user class (USRCLS) field in the user profile and the security level (QSECURITY) system value. If *USRCLS is specified, no additional special authorities can be specified for this user. If you specify *USRCLS when you create or change a user profile, the system puts the correct special authorities in the profile as if you had entered them. When you display profiles, you cannot tell whether special authorities were entered individually or entered by the system based on the user class. Table 4-1 on page 4-4 shows the default special authorities for each user class.
***NONE**	No special authority is granted to this user.
special-authority-name	Specify one or more special authorities for the user. The special authorities are described in the sections that follow.

***ALLOBJ Special Authority:** All-object (*ALLOBJ) special authority allows the user to access any resource on the system whether or not private authority exists for the user. Even if the user has *EXCLUDE authority to an object, *ALLOBJ special authority still allows the user to access the object.

Risks: *ALLOBJ special authority gives the user extensive authority over all resources on the system. The user can view, change, or delete any object. The user can also grant to other users the authority to use objects.

A user with *ALLOBJ authority cannot directly perform operations that require another special authority. For example, *ALLOBJ special authority does not allow a user to create another user profile, because creating user profiles requires *SECADM special authority. However, a user with *ALLOBJ special authority can submit a batch job to run using a profile that has the needed special authority. Giving *ALLOBJ special authority essentially gives a user access to all functions on the system.

***SECADM Special Authority:** Security administrator (*SECADM) special authority allows a user to create, change, and delete user profiles.

In addition, *SECADM special authority gives the user comprehensive authority to manage OfficeVision/400 objects and users. A user with *SECADM special authority can:

- Add users to the system distribution directory. This includes the right to create and change user profiles for OfficeVision/400 users.

- Display authority for documents or folders.
- Add and remove access codes to the system.
- Give and remove a user's access code authority
- Give and remove permission for users to work on another user's behalf
- Delete documents and folders.
- Delete document lists.
- Change distribution lists created by other users.

Only a user with *SECADM and *ALLOBJ special authority can give *SECADM special authority to another user.

OfficeVision/400 Administrator: The OfficeVision/400 licensed program allows you to give an administrator full or limited *SECADM special authority. An administrator who has full *SECADM special authority is able to work with system objects, such as libraries, while using the OfficeVision/400 program. An administrator with limited *SECADM special authority cannot work with system objects while using the OfficeVision/400 program.

The *Managing OfficeVision/400** manual provides more information about the OfficeVision/400 administrator authority.

Risks: The OfficeVision/400 user whose *SECADM special authority is not limited is able to work with user profiles, system values, network attributes, and other system objects. All of these have a major impact on the security and performance of your system. *SECADM special authority also gives the user comprehensive authority over OfficeVision/400 users and objects.

***JOBCTL Special Authority:** Job control (*JOBCTL) special authority allows the user to:

- Change, delete, hold, and release all files on any output queues specified as OPRCTL(*YES).
- Display, send, and copy all files on any output queues specified as DSPDTA(*YES or *NO) and OPRCTL(*YES).
- Hold, release, and clear job queues specified as OPRCTL(*YES).
- Hold, release, and clear output queues specified as OPRCTL(*YES).
- Hold, release, change, and cancel other users' jobs.
- Start writers, if the output queue is specified as OPRCTL(*YES).
- Change the running attributes of a job, such as the printer for a job.
- Stop subsystems.
- Perform an initial program load (IPL).

Securing printer output and output queues is discussed in "Security and Printing" on page 6-6.

You can change the job priority (JOBPTY) and the output priority (OUTPTY) of your own job without job control special authority. You must have *JOBCTL special authority to change the run priority (RUNPTY) of your own job.

Changes to the output priority and job priority of a job are limited by the priority limit (PTYLMT) in the profile of the user making the change.

Risks: A user with *JOBCTL special authority can change the priority of jobs and of printing, end a job before it has finished, or delete output before it has printed. *JOBCTL special authority can also give a user access to confidential spooled output, if output queues are specified OPRCTL(*YES). A user who abuses *JOBCTL special authority can cause negative impacts on individual jobs and on overall system performance.

***SPLCTL Special Authority:** Spool control (*SPLCTL) special authority allows the user to perform all spool control functions, such as changing, deleting, displaying, holding and releasing spooled files. The user can perform these functions on all output queues, regardless of any authorities for the output queue or the OPRCTL parameter for the output queue.

*SPLCTL special authority also allows the user to manage jobs on job queues, including canceling the jobs and changing their priorities. The user can perform these functions on all job queues, regardless of any authorities for the job queue or the OPRCTL parameter for the job queue.

Risks: The user with *SPLCTL special authority can perform any operation on any spooled file in the system. Confidential spooled files cannot be protected from a user with *SPLCTL special authority. The user with *SPLCTL special authority can also control jobs waiting in job queues. The user could run jobs out of sequence or cancel jobs that update critical files.

***SAVSYS Special Authority:** Save system (*SAVSYS) special authority gives the user the authority to save, restore, and free storage for all objects on the system, whether or not the user has object existence authority to the objects.

Risks: The user with *SAVSYS special authority can:

- Save an object and take it to another AS/400 system to be restored.
- Save an object and display the tape to view the data.
- Save an object and free storage, thus deleting the data portion of the object.
- Save a document and delete it.

***SERVICE Special Authority:** Service (*SERVICE) special authority allows the user to perform the display and alter service functions. The dump function can be performed without *SERVICE authority.

Risks: A user with *SERVICE special authority can display and change confidential information using service functions.

***AUDIT Special Authority:** Audit (*AUDIT) special authority gives the user the ability to change auditing characteristics. The user can:

- Change the system values that control auditing.
- Use the CHGOBJAUD and CHGDLOAUD commands to change auditing for objects.
- Use the CHGUSRAUD command to change auditing for a user.

Risks: A user with *AUDIT special authority can stop and start auditing on the system or prevent auditing of particular actions. If having an audit record of security-relevant events is important for your system, carefully control and monitor the use of *AUDIT special authority.

Recommendations for Special Authorities: Giving special authorities to users represents a security exposure. For each user, carefully evaluate the need for any special authorities. Keep track of which users have special authorities and periodically review their requirement for the authority.

In addition, control whether user profiles with special authorities can be used to submit jobs and whether programs run using their authority (adopted authority).

Special Environment

Add User prompt: Not shown

CL parameter: SPCENV

Length: 10

Special environment determines the environment the user operates in after signing on. The user can operate in the AS/400, the System/36, or the System/38 environment. When the user signs on, the system uses the routing program and the special environment in the user's profile to determine the user's environment. See Figure 4-3.

Possible Values for SPCENV:

***SYSVAL**	The QSPCENV system value is used to determine the environment when the user signs on, if the user's routing program is QCMD.
***NONE**	The user operates in the AS/400 environment.
***S36**	The user operates in the System/36 environment if the user's routing program is QCMD.

Recommendations: If the user runs a combination of AS/400 and System/36 applications, use the Start System/36 (STRS36) command before running System/36 applications rather than specifying the System/36 environment in the user profile. This provides better performance for the AS/400 applications.

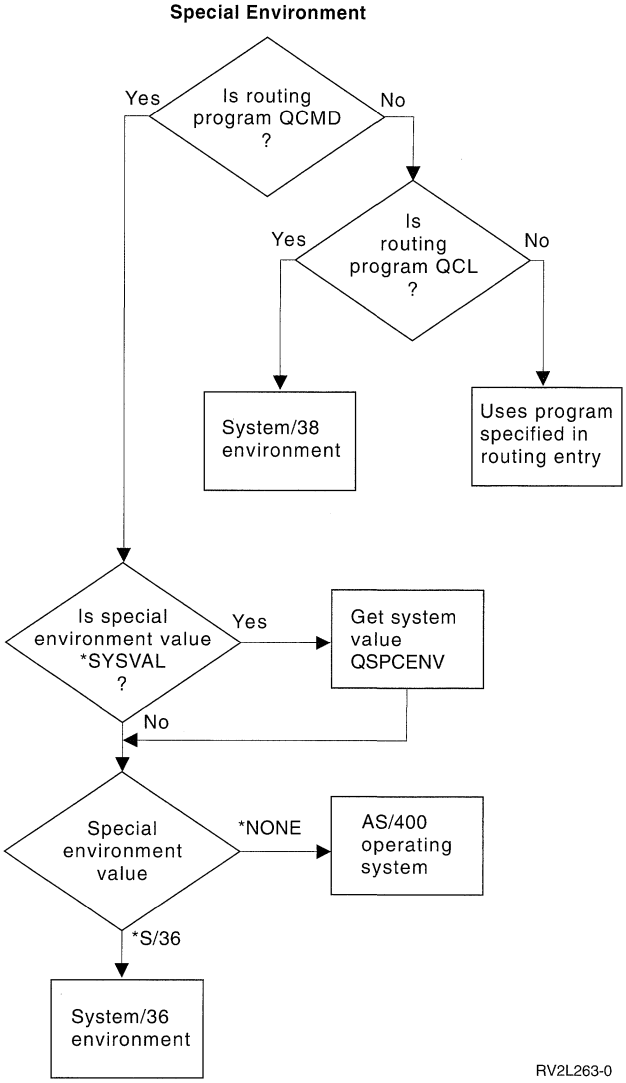


Figure 4-3. Determining the Special Environment

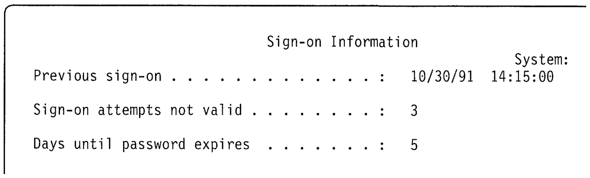
Display Sign-On Information

Add User prompt: Not shown

CL parameter: DSPSGNINF

Length: 7

The *Display sign-on information* field specifies whether the Sign-on Information display is shown when the user signs on. Figure 4-4 shows the display. Password expiration information is only shown if the password expires within seven days.



```
                          Sign-on Information
                                                                System:
 Previous sign-on . . . . . . . . . . . . . :   10/30/91  14:15:00

 Sign-on attempts not valid . . . . . . . . :   3

 Days until password expires  . . . . . . . :   5
```

Figure 4-4. Sign-On Information Display

Possible Values for DSPSGNINF:

***SYSVAL**	The QDSPSGNINF system value is used.
***NO**	The Sign-on Information display is not shown when the user signs on.
***YES**	The Sign-on Information display is shown when the user signs on.

Recommendations: The Sign-on Information display is a tool for users to monitor their profiles and to detect attempted misuse. Having all users see this display is recommended. Users with special authority or authority to critical objects should be encouraged to use the display to make sure no one attempts to use their profiles.

Password Expiration Interval

Add User prompt:	Not shown
CL parameter:	PWDEXPITV
Length:	5,0

Requiring users to change their passwords after a specified length of time reduces the risk of an unauthorized person accessing the system. The password expiration interval controls the number of days that a valid password can be used before it must be changed.

When a user's password has expired, the user receives a message at sign-on. The user can either press the Enter key to assign a new password or press F3 (Exit) to cancel the sign-on attempt without assigning a new password. If the user chooses to change the password, the Change Password display is shown and full password validation is run for the new password. Figure 4-2 on page 4-4 shows an example of the password expiration message.

Possible Values for PWDEXPITV:

***SYSVAL**	The QPWDEXPITV system value is used.
***NOMAX**	The system does not require the user to change the password.
password-expiration-interval	Specify a number from 1 through 366.

Recommendations: Set the QPWDEXPITV system value for an appropriate interval, such as 60 to 90 days. Use the *Password expiration interval* field in the user profile for individual users who should change their passwords more frequently, such as security administrators.

Limit Device Sessions

Add User prompt:	Not shown
CL parameter:	LMTDEVSSN
Length:	7

The *Limit device sessions* field controls whether a user can be signed on at more than one workstation at a time. The value does not restrict the use of the System Request menu or a second sign-on from the same device.

Possible Values for LMTDEVSSN:

***SYSVAL**	The QLMTDEVSSN system value is used.
***NO**	The user may be signed on to more than one device at the same time.
***YES**	The user may not be signed on to more than one device at the same time.

Recommendations: Limiting users to one workstation at a time is one way to discourage sharing passwords. Set the QLMTDEVSSN system value to 1 (YES). If some users have a requirement to sign on at multiple workstations, use the *Limit device sessions* field in the user profile for those users.

Keyboard Buffering

Add User prompt:	Not shown
CL parameter:	KBDBUF
Length:	10

This parameter specifies the keyboard buffering value used when a job is initialized for this user profile. The new value takes effect the next time the user signs on.

The keyboard buffering field controls two functions:

Type-ahead:
Lets the user type data faster than it can be sent to the system.

Attention key buffering:
If attention key buffering is on, the Attention key is treated like any other key. If attention key buffering is not on, pressing the Attention key results in sending the information to the system even when other workstation input is inhibited.

Possible Values for KBDBUF:

***SYSVAL**	The QKBDBUF system value is used.
***NO**	The type-ahead feature and Attention-key buffering option are not active for this user profile.
***TYPEAHEAD**	The type-ahead feature is active for this user profile.
***YES**	The type-ahead feature and Attention-key buffering option are active for this user profile.

Maximum Storage

Add User prompt:	Not shown
CL parameter:	MAXSTG
Length:	11,0

You can specify the maximum amount of auxiliary storage that is used to store permanent objects that are owned by a user profile, including objects placed in the temporary library

(QTEMP) during a job. Maximum storage is specified in kilobytes (1024 bytes).

If the storage needed is greater than the maximum amount specified when the user attempts to create an object, the object is not created.

When planning maximum storage for user profiles, consider the following system functions, which can affect the maximum storage needed by a user:

- A restore operation first assigns the storage to the user doing the restore operation, and then transfers the objects to the owner. Users who do large restore operations should have MAXSTG(*NOMAX) in their user profiles.
- The user profile that owns a journal receiver is assigned the storage as the receiver size grows. If new receivers are created, the storage continues to be assigned to the user profile that owns the active journal receiver. Users who own active journal receivers should have MAXSTG(*NOMAX) in their user profiles.
- If a user profile specifies OWNER(*GRPPRF), ownership of any object created by the user is transferred to the group profile after the object is created. However, the user creating the object must have adequate storage to contain any created object before the object ownership is transferred to the group profile.
- The owner of a library is assigned the storage for the descriptions of the objects that are placed in a library, even when the objects are owned by another user profile. Examples of such descriptions are text and program references.
- Storage is assigned to the user profile for temporary objects that are used during the processing of a job. Examples of such objects are commitment control blocks, file editing spaces, and documents.

Possible Values for MAXSTG:

***NOMAX**	As much storage as required can be assigned to this profile.
maximum-KB	Specify the maximum amount of storage in kilobytes (1 kilobytes equals 1024 bytes) that can be assigned to this user profile.

Priority Limit

Add User prompt: Not shown

CL parameter: PTYLMT

Length: 1

A batch job has three different priority values:

Run priority:
Determines how the job competes for machine resources when the job is running. Run priority is determined by the job's class.

Job priority:
Determines the scheduling priority for a batch job when the job is on the job queue. Job priority can be set by the job description or on the submit command.

Output priority:
Determines the scheduling priority for any output created by the job on the output queue. Output priority can be set by the job description or on the submit command.

The priority limit in the user profile determines the maximum scheduling priorities (job priority and output priority) allowed for any jobs the user submits. It controls priority when the job is submitted, as well as any changes made to priority while the job is running or waiting in a queue.

The priority limit also limits changes that a user with *JOBCTL special authority can make to another user's job. You cannot give someone else's job a higher priority than the limit specified in your own user profile.

If a batch job runs under a different user profile than the user submitting the job, the priority limits for the batch job are determined by the profile the job runs under. If a requested scheduling priority on a submitted job is higher than the priority limit in the user profile, the priority of the job is reduced to the level permitted by the user profile.

Possible Values for PTYLMT:

3	The default priority limit for user profiles is 3. The default priority for both job priority and output priority on job descriptions is 5. Setting the priority limit for the user profile at 3 gives the user the ability to move some jobs ahead of others on the queues.
priority-limit	Specify a value, 1 through 9. The highest priority is 1; the lowest priority is 9.

Recommendations: Using the priority values in job descriptions and on the submit job commands is usually a better way to manage the use of system resources than changing the priority limit in user profiles.

Use the priority limit in the user profile to control changes that users can make to submitted jobs. For example, system operators may need a higher priority limit so that they can move jobs in the queues.

Job Description

Add User prompt: Not shown

CL parameter: JOBD

Length 10 (job description name)
10 (library name)

Authority: *USE for job description
*READ for library

When a user signs on, the system looks at the workstation entry in the subsystem description to determine what job description to use for the interactive job. If the workstation entry specifies *USRPRF for the job description, the job description in the user profile is used.

The job description for a batch job is specified when the job is started. It can be specified by name, or it can be the job description from the user profile under which the job runs.

A job description contains a specific set of job-related attributes, such as which job queue to use, scheduling priority, routing data, message queue severity, library list and output information. The attributes determine how each job is run on the system.

See the *Work Management Guide* for more information about job descriptions and their uses.

Possible Values for JOBD:

QDFTJOBD	The system-supplied job description found in library QGPL is used. You can use the Display Job Description (DSPJOBD) command to see the attributes contained in this job description.
job-description-name	Specify the name of the job description, 10 characters or less.

Possible Values for JOBD Library:

***LIBL**	The library list is used to locate the job description.
***CURLIB**	The current library for the job is used to locate the job description. If no current library entry exists in the library list, QGPL is used.
library-name	Specify the library where the job description is located, 10 characters or less.

Recommendations: For interactive jobs, the job description is a good method of controlling library access. You can use a job description for an individual to specify a unique library list, rather than using the QUSRLIBL system value.

Group Profile

Add User prompt:	User Group
CL parameter:	GRPPRF
Length:	10
Authority:	To specify a group when creating or changing a user profile, you must have *OBJMGT and *CHANGE authority to the group profile.

Note: Adopted authority is not used to check for *OBJMGT authority to the group profile. For more information about adopted authority, see “Objects That Adopt the Owner's Authority” on page 5-6.

Specifying a group profile name makes the user a member of the group profile. The group profile can provide the user with authority to use objects for which the user does not have specific authority.

When a group profile is specified in a user profile, the user is automatically granted *OBJMGT and *CHANGE authorities to the group profile.

See “Planning Group Profiles” on page 7-10 for more information about using group profiles.

Possible Values for GRPPRF:

***NONE**	No group profile is used with this user profile.
user-profile-name	Specify the name of a group profile of which this user profile is a member.

Owner

Add User prompt:	Not shown
CL parameter:	OWNER
Length:	10

If the user is a member of a group, you can specify whether the user profile or the group profile is the owner of any objects created by this user. You can only specify the *Owner* field if you have specified the *Group profile* field.

Possible Values for OWNER:

***USRPRF**	This user profile is the owner of any new objects it creates.
***GRPPRF**	The group profile is made the owner of any objects created by the user and is given all (*ALL) authority to the objects. The user profile is not given any specific authority to new objects it creates. If *GRPPRF is specified, you must specify a group profile name in the GRPPRF parameter, and the GRPAUT parameter must be *NONE.

Group Authority

Add User prompt:	Not shown
CL parameter:	GRPAUT
Length:	10

If the user profile is a member of a group and OWNER(*USRPRF) is specified, the *Group authority* field controls what authority is given to the group profile for any objects created by this user.

Group authority can be specified only when GRPPRF is not *NONE and OWNER is *USRPRF.

Possible Values for GRPAUT: [1]

***NONE**	No specific authority is given to the group profile when this user creates objects.
***ALL**	The group profile is given all management and data authorities to any new objects the user creates.
***CHANGE**	The group profile is given the authority to change any objects the user creates.
***USE**	The group profile is given authority to view any objects the user creates.

Possible Values for GRPAUT: [1]

***EXCLUDE**	The group profile is specifically denied access to any new objects created by the user.

[1] See "Defining How Information Can Be Accessed" on page 5-2 for a complete explanation of the authorities that can be granted.

Accounting Code

Add User prompt:	Not shown
CL parameter:	ACGCDE
Length:	15

Job accounting is an optional function used to gather information about the use of system resources. The accounting level (QACGLVL) system value determines whether job accounting is active. The accounting code for a job comes from either the job description or the user profile. The accounting code can also be specified when a job is running using the Change Accounting Code (CHGACGCDE) command.

See the *Work Management Guide* for more information about job accounting.

Possible Values for ACGCDE:

***BLANK**	An accounting code of 15 blanks is assigned to this user profile.
accounting-code	Specify a 15-character accounting code. If less than 15 characters are specified, the string is padded with blanks on the right.

Document Password

Add User prompt:	Not shown
CL parameter:	DOCPWD
Length:	8

You can specify a document password for the user to protect the distribution of personal mail from being viewed by people working on behalf of the user. The document password is supported by some Document Interchange Architecture (DIA) products, such as the Displaywriter.

See the *Planning For and Setting Up OfficeVision/400** manual for more information about using a document password and other methods for protecting documents.

Possible Values for DOCPWD:

***NONE**	No document password is used by this user.
document-password	Specify a document password for this user. The password must consist of from 1 through 8 characters (letters A through Z and numbers 0 through 9). The first character of the document password must be alphabetic; the remaining characters can be alphanumeric. Embedded blanks, leading blanks, and special characters are not allowed.

Message Queue

Add User prompt:	Not shown
CL parameter:	MSGQ
Length:	10 (message queue name) 10 (library name)
Authority:	*USE for message queue, if it exists. *READ for library, if the message queue exists. *ADD for library, if the message queue does not exist.

You can specify the name of a message queue for a user. A **message queue** is an object on which messages are placed when they are sent to a person or a program. A message queue is used when a user sends or receives messages. If the message queue does not exist, it is created when the profile is created or changed. The message queue is owned by the profile being created or changed. The user creating the profile is given *ALL authority to the message queue.

If the message queue for a user profile is changed using the Change User Profile (CHGUSRPRF) command, the previous message queue is not automatically deleted by the system.

If a user profile is created with a password of *NONE, a message queue is not created.

For more information about message queues, see the *Operator's Guide*.

Possible Values for MSGQ:

***USRPRF**	A message queue with the same name as the user profile name is used as the message queue for this user. If the message queue does not exist, it is created in library QUSRSYS.
message-queue-name	Specify the message queue name that is used for this user. If you specify a message queue name, you must specify the library parameter.

Possible Values for MSGQ Library:

***LIBL**	The library list is used to locate the message queue. If the message queue does not exist, you cannot specify *LIBL.
***CURLIB**	The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used. If the message queue does not exist, it is created in the current library or QGPL.
library-name	Specify the library where the message queue is located. If the message queue does not exist, it is created in this library.

Recommendations: When a user signs on, the message queue in the user profile is allocated to that user's job. If the message queue is already allocated to another job, the user receives a warning message during sign-on. To avoid this, give each user profile a unique message queue, preferably with the same name as the user profile.

Delivery

Add User prompt:	Not shown
CL parameter:	DLVRY
Length:	10
Authority:	*USE for message queue *READ for library

The delivery mode of a message queue determines whether the user is interrupted when a new message arrives on the queue. The delivery mode specified in the user profile applies to the user's personal message queue. If you change the message queue delivery in the user profile and the user is signed on, the change takes affect the next time the user signs on. You can also change the delivery of a message queue with the Change Message Queue (CHGMSGQ) command.

Possible Values for DLVRY:

***NOTIFY** The job that the message queue is assigned to is notified when a message arrives at the message queue. For interactive jobs at a workstation, the audible alarm is sounded and the message-waiting light is turned on. The type of delivery cannot be changed to *NOTIFY if the message queue is also being used by another user.

***BREAK** The job that the message queue is assigned to is interrupted when a message arrives at the message queue. If the job is an interactive job, the audible alarm is sounded (if the alarm is installed). The type of delivery cannot be changed to *BREAK if the message queue is also being used by another user.

***HOLD** The messages are held in the message queue until they are requested by the user or program.

***DFT** Messages requiring replies are answered with their default reply; information-only messages are ignored.

Severity

Add User prompt:	Not shown
CL parameter:	SEV
Length:	2,0
Authority:	*USE for message queue *READ for library

If a message queue is in *BREAK or *NOTIFY mode, the severity code determines the lowest-level messages that are delivered to the user. Messages whose severity is lower than the specified severity code are held in the message queue without the user being notified.

If you change the message queue severity in the user profile and the user is signed on, the change takes effect the next time the user signs on. You can also change the severity of a message queue with the CHGMSGQ command.

Possible Values for SEV:

00: If a severity code is not specified, 00 is used. The user is notified of all messages, if the message queue is in *NOTIFY or *BREAK mode.

severity-code Specify a value, 00 through 99, for the lowest severity code that causes the user to be notified. Any 2-digit value can be specified, even if no severity code has been defined for it (either defined by the system or by the user).

Print Device

Add User prompt:	Default printer
CL parameter:	PRTDEV
Length:	10

You can specify the printer used to print the output for this user. Spooled files are placed on an output queue with the same name as the printer when the output queue (OUTQ) is specified as the print device (*DEV).

The print device and output queue information from the user profile are used only if the printer file specifies *JOB and the job description specifies *USRPRF. For more information about directing printer output, see the *Guide to Programming for Printing* manual.

Possible Values for PRTDEV:

***WRKSTN** The printer assigned to the user's workstation (in the device description) is used.

***SYSVAL** The default system printer specified in the QPRTDEV system value is used.

print-device-name Specify the name of the printer that is used to print the output for this user.

Output Queue

Add User prompt:	Not shown
CL parameter:	OUTQ
Length:	10 (output queue name) 10 (library name)
Authority:	*USE for output queue *READ for library

Both interactive and batch processing may result in spooled files that are to be sent to a printer. Spooled files are placed on an output queue. The system can have many different output queues. An output queue does not have to be attached to a printer to receive new spooled files.

The print device and output queue information from the user profile are used only if the printer file specifies *JOB and the job description specifies *USRPRF. For more information about directing printer output, see the *Guide to Programming for Printing* manual.

Possible Values for OUTQ:

*WRKSTN	The output queue assigned to the user's workstation (in the device description) is used.
*DEV	An output queue with the same name as the print device specified on the PRTDEV parameter is used.
output-queue-name	Specify the name of the output queue that is to be used. The output queue must already exist. If an output queue is specified, the library must be specified also.

Possible Values for OUTQ library:

*LIBL	The library list is used to locate the output queue.
*CURLIB	The current library for the job is used to locate the output queue. If no current library entry exists in the library list, QGPL is used.
library-name	Specify the library where the output queue is located.

Attention-Key-Handling Program

Add User prompt:	Not shown
CL parameter:	ATNPGM
Length:	10 (program name) 10 (library name)
Authority:	*USE for program *READ for library

The **Attention-key-handling program** (ATNPGM) is the program that is called when the user presses the Attention (ATTN) key during an interactive job.

The ATNPGM is activated only if the user's routing program is QCMD. The ATNPGM is activated before the initial program is called. If the initial program changes the ATNPGM, the new ATNPGM remains active only until the initial program ends. If the Set Attention-Key-Handling Program (SETATNPGM) command is run from a command line or an application, the new ATNPGM specified overrides the ATNPGM from the user profile.

Note: See "Starting an Interactive Job" on page 6-1 for more information about the processing sequence when a user signs on.

The *Limit capabilities* field determines if a different Attention-key-handling program can be specified by the user with the Change Profile (CHGPRF) command.

Possible Values for ATNPGM:

*SYSVAL	The QATNPGM system value is used.
*NONE	No Attention-key-handling program is used by this user.
*ASSIST	Operational Assistant (QEZMAIN) is used.
program-name	Specify the name of the Attention-key-handling program. If a program name is specified, a library must be specified.

Possible Values for ATNPGM Library:

*LIBL	The library list is used to locate the Attention-key-handling program.
*CURLIB	The current library for the job is used to locate the Attention-key-handling program. If no current library entry exists in the library list, QGPL is used.
library-name:	Specify the library where the Attention-key-handling program is located.

Sort Sequence

Add User prompt:	Not shown
CL parameter:	SRTSEQ
Length:	10 (value or table name) 10 (library name)
Authority:	*USE for table *READ for library

You can specify what sort sequence is used for this user's output. You can use system-provided sort tables or create your own. A sort table may be associated with a particular language identifier on the system. The *National Language Support Planning Guide* provides more information about using sort sequences.

Possible Values for SRTSEQ:

*SYSVAL	The QSRTSEQ system value is used.
*HEX	The standard hexadecimal sort sequence is used for this user.
*LANGIDSHR	The sort sequence table associated with the user's language identifier is used. The table can contain the same weight for multiple characters.
*LANGIDUNQ	The sort sequence table associated with the user's language identifier is used. The table must contain a unique weight for each character in the code page.
table-name	Specify the name of the sort sequence table for this user.

Possible Values for SRTSEQ Library:

*LIBL	The library list is used to locate the table specified for the SRTSEQ value.
*CURLIB	The current library for the job is used to locate the table specified for the SRTSEQ value. If no current library entry exists in the library list, QGPL is used.
library-name	Specify the library where the sort sequence table is located.

Language Identifier

Add User prompt:	Not shown
CL parameter:	LANGID
Length:	10

You can specify the language identifier to be used by the system for the user. To see a list of language identifiers, press F4 (prompt) on the language identifier parameter from the Create User Profile display or the Change User Profile display.

Possible Values for LANGID:

***SYSVAL:**	The system value QLANGID is used to determine the language identifier.
language-identifier	Specify the language identifier for this user.

Country Identifier

Add User prompt: Not shown

CL parameter: CNTRYID

Length: 10

You can specify the country identifier to be used by the system for the user. To see a list of country identifiers, press F4 (prompt) on the country identifier parameter from the Create User Profile display or the Change User Profile display.

Possible Values for CNTRYID:

***SYSVAL**	The system value QCNTRYID is used to determine the country identifier.
country-identifier	Specify the country identifier for this user.

Coded Character Set Identifier

Add User prompt: Not shown

CL parameter: CCSID

Length: 5,0

You can specify the coded character set identifier to be used by the system for the user. To see a list of coded character set identifiers, press F4 (prompt) on the coded character set identifier parameter from the Create User Profile display or the Change User Profile display.

Possible Values for CCSID:

***SYSVAL**	The QCCSID system value is used to determine the coded character set identifier.
coded-character-set-identifier	Specify the coded character set identifier for this user.

User Options

Add User prompt: Not shown

CL parameter: USROPT

Length: 240 (10 characters each)

The *User options* field allows you to customize certain system displays and functions for the user. You can specify multiple values for the user option parameter.

Possible Values for USROPT:

***NONE**	No special options are used for this user. The standard system interface is used.
***CLKWD**	Keywords are shown instead of the possible parameter values when a control language (CL) command is prompted. This is equivalent to pressing F11 from the normal control language (CL) command prompting display.
***EXPERT**	When the user views displays that show object authority, such as the Edit Object Authority display or the Edit Authorization List Display, detailed authority information is shown without the user having to press F11 (Display detail).
***HLPFULL**	The user sees full display help information instead of a window.
***PRTMSG**	A message is sent to the user's message queue when a spooled file is printed for this user.
***ROLLKEY**	The actions of the Page Up and Page Down keys are reversed.
***NOSTSMSG**	Status messages usually shown at the bottom of the display are not shown to the user.
***STSMSG**	Status messages are displayed when sent to the user.

Authority

Add User prompt: Not shown

CL parameter: AUT

Length: 10

The *Authority* field specifies the public authority to the user profile. The authority to a profile controls many functions associated with the profile, such as:

Changing it
Displaying it
Deleting it
Submitting a job using it
Specifying it in a job description
Transferring object ownership to it
Adding members, if it is a group profile

Possible Values for AUT:

***EXCLUDE**	The public is specifically denied access to the user profile.
***ALL**	The public is given all management and data authorities to the user profile.
***CHANGE**	The public is given the authority to change the user profile.
***USE**	The public is given authority to view the user profile.

See “Defining How Information Can Be Accessed” on page 5-2 for a complete explanation of the authorities that can be granted.

Recommendations: To prevent misuse of user profiles that have authority to critical objects, make sure the public authority to the profiles is *EXCLUDE. Possible misuses of a profile include submitting a job that runs under that user profile or changing a program to adopt the authority of that user profile.

Object Auditing

Add User prompt: Not shown

CL parameter: OBJAUD

Length: 10

The object auditing value for a user profile works with the object auditing value for an object to determine whether the user's access of an object is audited. Object auditing for a user profile cannot be specified on any user profile displays. Use the CHGUSRAUD command to specify object auditing for a user. Only a user with *AUDIT special authority can use the CHGUSRAUD command.

Possible Values for OBJAUD:

***NONE** The OBJAUD value for objects determines whether object auditing is done for this user.

***CHANGE** If the OBJAUD value for an object specifies *USRPRF, an audit record is written when this user changes the object.

***ALL** If the OBJAUD value for an object specifies *USRPRF, an audit record is written when this user changes or reads the object.

Table 4-4 shows how the OBJAUD values for the user and the object work together:

Table 4-4. Auditing Performed for Object Access

OBJAUD Value for Object	**OBJAUD Value for User**		
	*NONE	*CHANGE	*ALL
*NONE	None	None	None
*USRPRF	None	Change	Change and Use
*CHANGE	Change	Change	Change
*ALL	Change and Use	Change and Use	Change and Use

"Planning the Auditing of Object Access" on page 9-9 provides information about how to use system values and the object auditing values for users and objects to meet your security auditing needs.

Action Auditing

Add User prompt: Not shown

CL parameter: AUDLVL

Length: 640

For an individual user, you can specify which security-relevant actions should be recorded in the audit journal. The actions specified for an individual user apply in addition to the actions specified for all users by the QAUDLVL system value. Action auditing for a user profile cannot be specified on any user profile displays. It is defined using the CHGUSRAUD command. Only a user with *AUDIT special authority can use the CHGUSRAUD command.

Possible Values for AUDLVL:

***NONE** The QAUDLVL system value controls action auditing for this user. No additional auditing is done.

***CMD** Command strings are logged. *CMD can be specified only for individual users. Command string auditing is not available as a system-wide option using the QAUDLVL system value.

***CREATE** Object create operations are logged.

***DELETE** Object delete operations are logged.

***JOBDTA** Job changes are logged.

***OBJMGT** Object move and rename operations are logged.

***OFCSRV** Changes to the system distribution directory and office mail actions are logged.

***PGMADP** Obtaining authority to an object through a program that adopts authority is logged.

***SAVRST** Restore operations are logged.

***SECURITY** Security-related functions are logged.

***SERVICE** Using service tools is logged.

***SPLFDTA** Actions performed on spooled files are logged.

***SYSMGT** Use of system management functions is logged.

"Planning the Auditing of Actions" on page 9-4 provides information about how to use system values and the action auditing for users to meet your security auditing needs.

Additional Information Associated with a User Profile

The previous sections described the fields you specify when you create and change user profiles. Other information is associated with a user profile on the system and saved with it:

- Private authorities
- Owned object information

The amount of this information affects the time it takes to save and restore profiles and to build authority displays. "How Security Information Is Stored" on page 8-1 provides more information about how user profiles are stored and saved.

Private Authorities: All the private authorities a user has to objects are stored with the user profile. When a user needs authority to an object, the user's private authorities may be searched. "Flowchart 2: How User Authority to an Object Is Checked" on page 5-12 provides more information about authority checking.

You can display a user's private authorities using the Display User Profile command: `DSPUSRPRF` *`user-profile-name`* `TYPE(*OBJAUT)`. To change a user's private authorities, you use the commands that work with object authorities, such as Edit Object Authority (EDTOBJAUT).

You can copy all the private authorities from one user profile to another using the Grant User Authority (GRTUSRAUT) command. See "Copying Authority from a User" on page 5-27 for more information.

Owned Object Information: Private authority information for an object is also stored with the user profile that owns the object. This information is used to build system displays that work with object authority. If a profile owns a large number of objects that have many private authorities, the performance of building object authority displays for these objects can be affected.

Working with User Profiles

This part of the chapter describes the commands and displays you use to create, change, and delete user profiles. All the fields, options, and function keys are not described. Use online information for details.

You must have *SECADM special authority to work with user profiles.

Creating User Profiles

You can create user profiles in several ways:

- Using the Work with User Profiles (WRKUSRPRF) list display.
- Using the Create User Profile (CRTUSRPRF) command.
- Using the Work with User Enrollment option from the SETUP menu.

Figure 4-1 on page 4-1 illustrates these methods.

Using the Work with User Profiles Command: You can enter a specific profile name, a generic profile set, or *ALL on the WRKUSRPRF command. The assistance level determines which list display you see:

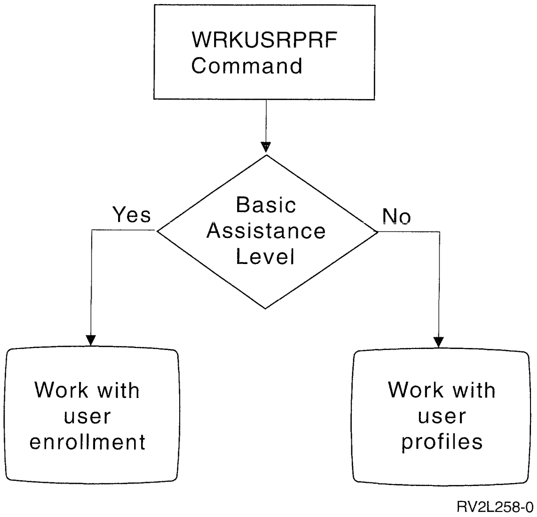
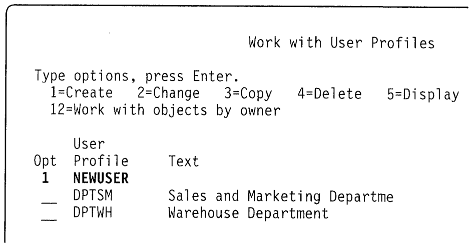


Figure 4-5. Assistance Level for User Profile Displays

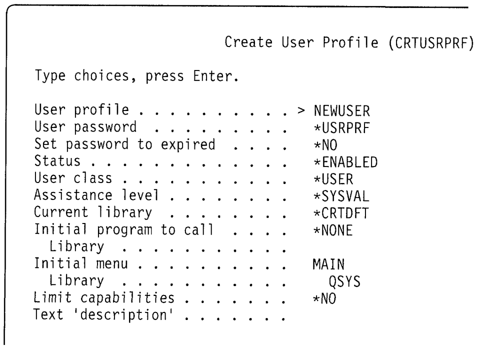
You can specify the ASTLVL (assistance level) parameter on the command. If you do not specify ASTLVL, the system uses the assistance level stored with your user profile.

On the Work with User Profiles display, type 1 and the name of the profile you want to create:



```
                        Work with User Profiles

Type options, press Enter.
  1=Create   2=Change   3=Copy   4=Delete   5=Display
  12=Work with objects by owner

     User
Opt  Profile     Text
 1   NEWUSER
 _   DPTSM       Sales and Marketing Departme
 _   DPTWH       Warehouse Department
```

You see the Create User Profile display:



```
                       Create User Profile (CRTUSRPRF)

Type choices, press Enter.

User profile . . . . . . . . . . > NEWUSER
User password  . . . . . . . . .   *USRPRF
Set password to expired  . . . .   *NO
Status . . . . . . . . . . . . .   *ENABLED
User class . . . . . . . . . . .   *USER
Assistance level . . . . . . . .   *SYSVAL
Current library  . . . . . . . .   *CRTDFT
Initial program to call  . . . .   *NONE
  Library  . . . . . . . . . . .
Initial menu . . . . . . . . . .   MAIN
  Library  . . . . . . . . . . .     QSYS
Limit capabilities . . . . . . .   *NO
Text 'description' . . . . . . .
```

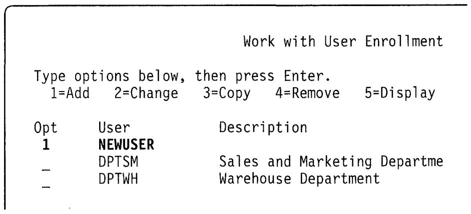
The Create User Profile display shows all the fields in the user profile. Use F10 (Additional parameters) and page down to enter more information. Use F11 (Display keywords) to see the parameter names.

The Create User Profile display does not enroll a user in the OfficeVision/400 licensed program or add the user to the system directory. The Add User display gives you the option to enroll the user in the OfficeVision/400 licensed program.

Using the Create User Profile Command: You can use the CRTUSRPRF command to create a user profile. You can enter parameters with the command, or you can request prompting (F4) and see the Create User Profile display.

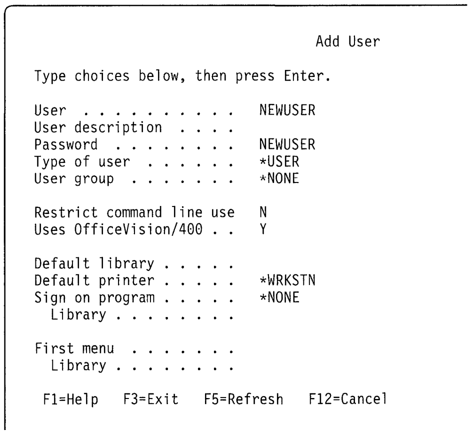
Using the Work with User Enrollment Option: Select the Work with User Enrollment option from the SETUP menu. The assistance level stored with your user profile determines whether you see the Work with User Profiles display or the Work with User Enrollment display. See Figure 4-5 on page 4-18. You can use F21 (Select assistance level) to change levels.

On the Work with User Enrollment display, use option 1 (Add) to add a new user to the system.



```
                     Work with User Enrollment

Type options below, then press Enter.
  1=Add   2=Change   3=Copy   4=Remove   5=Display

Opt     User           Description
 1      NEWUSER
 _      DPTSM          Sales and Marketing Departme
 _      DPTWH          Warehouse Department
```

You see the Add User display:

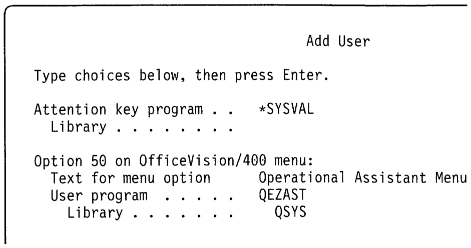


```
                              Add User

Type choices below, then press Enter.

User . . . . . . . . . .   NEWUSER
User description . . . .
Password . . . . . . . .   NEWUSER
Type of user . . . . . .   *USER
User group . . . . . . .   *NONE

Restrict command line use   N
Uses OfficeVision/400 . .   Y

Default library . . . . .
Default printer . . . . .   *WRKSTN
Sign on program . . . . .   *NONE
  Library . . . . . . . .

First menu . . . . . . .
  Library . . . . . . . .

 F1=Help   F3=Exit   F5=Refresh   F12=Cancel
```

The Add User display is designed for a security administrator without a technical background. It does not show all of the fields in the user profile. Default values are used for all fields that are not shown.

Note: If you use the Add User display, you are limited to eight-character user profile names.

Page down to see the second display:



```
                              Add User

Type choices below, then press Enter.

Attention key program . .   *SYSVAL
  Library . . . . . . . .

Option 50 on OfficeVision/400 menu:
  Text for menu option       Operational Assistant Menu
  User program . . . . .     QEZAST
    Library . . . . . . .      QSYS
```

The Add user display automatically adds an entry in the system directory with the same user ID as the user profile name (the first eight characters) and an address of the system name.

If you specify Y to the *Uses OfficeVision/400* prompt, the system creates the user profile and enrolls the user in the OfficeVision/400 licensed program. The system:

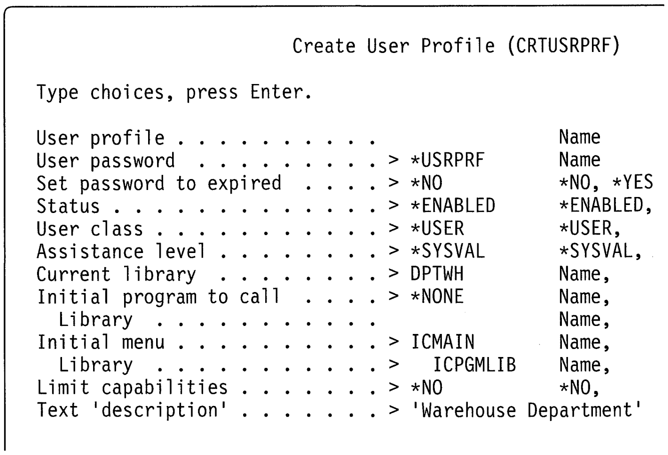
- Creates a calendar and a folder with the same name as the user profile.
- Sets up the program and text you specify for option 50 on the OfficeVision/400 menu.

Copying User Profiles

You can create a user profile by copying another user profile or a group profile. You may want to set up one profile in a group as a pattern. Copy the first profile in the group to create additional profiles.

You can copy a profile interactively from either the Work with User Enrollment display or the Work with User Profiles display. No command exists to copy a user profile.

Copying from the Work with User Profiles Display: On the Work with User Profiles display, type 3 in front of the profile you want to copy. You see the Create User Profile display:



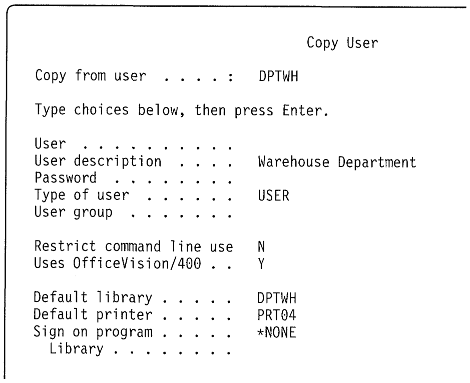
```
                   Create User Profile (CRTUSRPRF)

Type choices, press Enter.

User profile . . . . . . . . .                    Name
User password  . . . . . . . . > *USRPRF          Name
Set password to expired  . . . > *NO              *NO, *YES
Status . . . . . . . . . . . . > *ENABLED         *ENABLED,
User class . . . . . . . . . . > *USER            *USER,
Assistance level . . . . . . . > *SYSVAL          *SYSVAL,
Current library  . . . . . . . > DPTWH            Name,
Initial program to call  . . . > *NONE            Name,
  Library  . . . . . . . . . .                    Name,
Initial menu . . . . . . . . . > ICMAIN           Name,
  Library  . . . . . . . . . . >   ICPGMLIB       Name,
Limit capabilities . . . . . . > *NO              *NO,
Text 'description' . . . . . . > 'Warehouse Department'
```

All the values from the copy-from user profile are shown on the Create User Profile display, except these fields:

User profile	Blank. Must be filled in.
Password	*USRPRF
Message queue	*USRPRF
Document password	*NONE
Authority	*EXCLUDE

You can change any fields on the Create User Profile display. Private authorities of the copy-from profile are not copied.

Copying from the Work with User Enrollment Display: On the Work with User Enrollment display, type 3 in front of the profile you want to copy. You see the Copy User display:



```
                           Copy User

  Copy from user  . . . . :   DPTWH

  Type choices below, then press Enter.

  User  . . . . . . . . . .
  User description  . . . .   Warehouse Department
  Password  . . . . . . . .
  Type of user  . . . . . .   USER
  User group  . . . . . . .

  Restrict command line use   N
  Uses OfficeVision/400 . .   Y

  Default library . . . . .   DPTWH
  Default printer . . . . .   PRT04
  Sign on program . . . . .   *NONE
    Library . . . . . . . .
```

All values from the copy-from profile appear on the Add User display, except the following:

User	Blank. Must be filled in.
Password	Blank. If you do not enter a value, the profile is created with the password equal to the user profile name.

You can change any fields on the Copy User display. User profile fields that do not appear on the basic assistance level version are still copied from the copy-from profile, with the following exceptions:

Message queue	*USRPRF
Document password	*NONE
Authority	*EXCLUDE

Private authorities of the copy-from profile are not copied.

Copying Private Authorities: You can copy the private authorities from one user profile to another using the Grant User Authority (GRTUSRAUT) command. This can be useful in some situations, but should not be used in place of group profiles or authorization lists. Copying authorities does not help you manage similar authorities in the future, and it can cause performance problems on your system.

The topic "Copying Authority from a User" on page 5-27 has more information about using this command.

Changing User Profiles

You can change a user profile using option 2 (Change) from either the Work with User Profiles display or the Work with User Enrollment display. You can also use the Change User Profile (CHGUSRPRF) command.

Users who are allowed to enter commands can change some parameters of their own profiles using the Change Profile (CHGPRF) command.

Deleting User Profiles

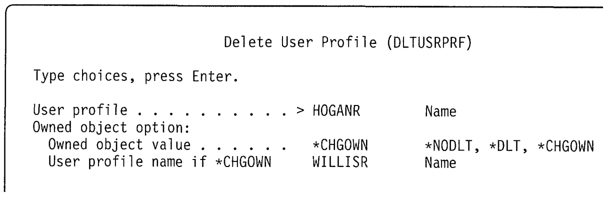
You cannot delete a user profile that owns objects. You must delete any objects owned by the profile or transfer ownership of those objects to another profile. Both basic assistance level and intermediate assistance level allow you to handle owned objects when you delete a profile.

When you delete a user profile, the user is removed from all distribution lists and from the system directory.

You do not need to change ownership of or delete the user's message queue. The system automatically deletes the message queue when the profile is deleted.

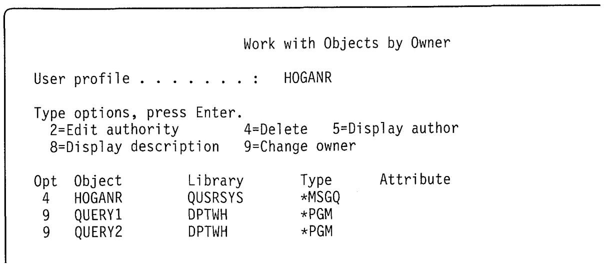
You cannot delete a group profile that has members. To list the members of a group profile, type DSPUSRPRF *group-profile-name* *GRPMBR. Change the GRPPRF field in each member profile before deleting the group profile.

Using the Delete User Profile Command: You can enter the Delete User Profile (DLTUSRPRF) command directly, or you can use option 4 (Delete) from the Work with User Profiles display. The DLTUSRPRF command has parameters allowing you to handle all objects owned by the profile:



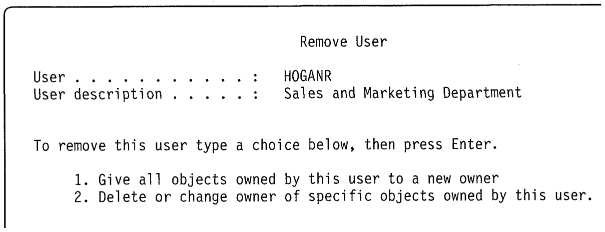
```
                        Delete User Profile (DLTUSRPRF)

  Type choices, press Enter.

  User profile . . . . . . . . . . > HOGANR        Name
  Owned object option:
    Owned object value . . . . . .   *CHGOWN       *NODLT, *DLT, *CHGOWN
    User profile name if *CHGOWN     WILLISR       Name
```

You can delete all the owned objects or transfer them to a new owner. If you want to handle owned objects individually, you can use the Work with Objects by Owner (WRKOBJOWN) command.



```
                      Work with Objects by Owner

User profile . . . . . . . :    HOGANR

Type options, press Enter.
  2=Edit authority         4=Delete   5=Display author
  8=Display description    9=Change owner

Opt  Object         Library        Type        Attribute
 4   HOGANR         QUSRSYS        *MSGQ
 9   QUERY1         DPTWH          *PGM
 9   QUERY2         DPTWH          *PGM
```

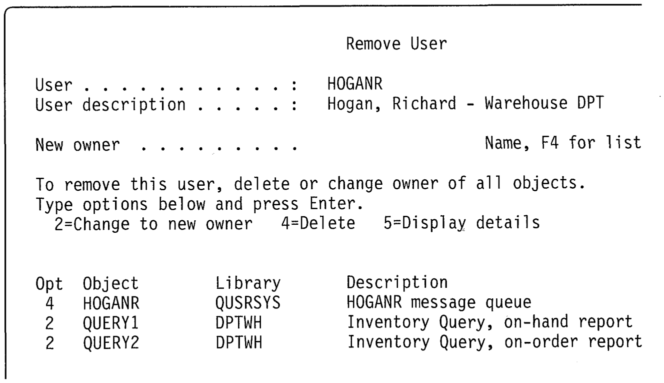
Using the Remove User Option: From the Work with User Enrollment display, type 4 (Remove) in front of the profile you want to delete. You see the Remove User display:



```
                                Remove User

User . . . . . . . . . . . :    HOGANR
User description . . . . . :    Sales and Marketing Department


To remove this user type a choice below, then press Enter.

     1. Give all objects owned by this user to a new owner
     2. Delete or change owner of specific objects owned by this user.
```

To change the ownership of all objects before deleting the profile, select option 1. You see a display prompting you for the new owner.

To handle the objects individually, select option 2. You see a detailed Remove User display:



```
                               Remove User

User . . . . . . . . . . . :    HOGANR
User description . . . . . :    Hogan, Richard - Warehouse DPT

New owner  . . . . . . . . .                       Name, F4 for list

To remove this user, delete or change owner of all objects.
Type options below and press Enter.
  2=Change to new owner   4=Delete   5=Display details


Opt  Object       Library      Description
 4   HOGANR       QUSRSYS      HOGANR message queue
 2   QUERY1       DPTWH        Inventory Query, on-hand report
 2   QUERY2       DPTWH        Inventory Query, on-order report
```

Use the options on the display to delete objects or transfer them to a new owner. When all objects have been removed from the display, you can delete the profile.

Notes:

1. You can use F13 to delete all the objects owned by the user profile.
2. Spooled files do not appear on the Work with Objects by Owner display. You can delete a user profile even though that profile still owns spooled files. After you have deleted a user profile, use the Work with Spooled Files (WRKSPLF) command to locate and delete any spooled files owned by the user profile, if they are no longer needed.

Enabling a User Profile

If the QMAXSIGN and QMAXSGNACN system values on your system are set up to disable a user profile after too many sign-on attempts, you may want someone like a system operator to enable the profile by changing the status to *ENABLE. However, to enable a user profile, you must have *SECADM special authority and *OBJMGT and *USE authority to the user profile. Normally, a system operator does not have *SECADM special authority.

A solution is to use a simple program which adopts authority:

1. Create a CL program owned by a user who has *SECADM special authority and *OBJMGT and *USE authority to the user profiles on the system. Adopt the authority of the owner when the program is created by specifying USRPRF(*OWNER).
2. Use the EDTOBJAUT command to make the public authority to the program *EXCLUDE and give the system operators *USE authority.
3. The operator enables the profile by entering:

   `CALL ENABLEPGM` *`profile-name`*
4. The main part of the ENABLEPGM program looks like this:

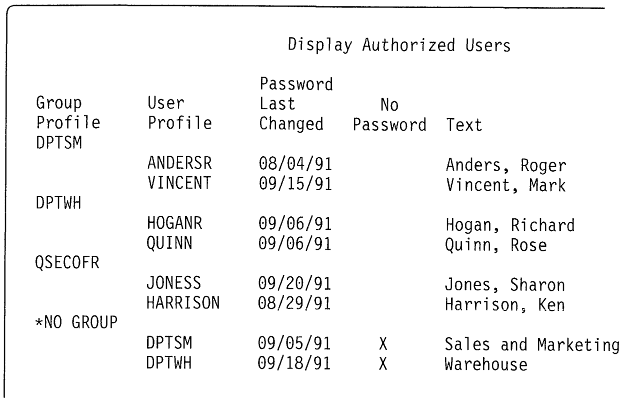
   ```
   PGM &PROFILE
   DCL VAR(&PROFILE) TYPE(*CHAR) LEN(10)
   CHGUSRPRF USRPRF(&PROFILE) STATUS(*ENABLED)
   ENDPGM
   ```

Listing User Profiles

You can display and print information about user profiles in a variety of formats.

Displaying an Individual Profile: To display the values for an individual user profile, use option 5 (Display) from either the Work with User Enrollment display or the Work with User Profiles display. Or, you can use the Display User Profile (DSPUSRPRF) command.

Listing All Profiles: Use the Display Authorized Users (DSPAUTUSR) command to either print or display all the user profiles on the system. The sequence (SEQ) parameter on the command allows you to sort the list either by profile name or by group profile.



```
                          Display Authorized Users

                           Password
Group         User         Last          No
Profile       Profile      Changed    Password   Text
DPTSM
              ANDERSR      08/04/91              Anders, Roger
              VINCENT      09/15/91              Vincent, Mark
DPTWH
              HOGANR       09/06/91              Hogan, Richard
              QUINN        09/06/91              Quinn, Rose
QSECOFR
              JONESS       09/20/91              Jones, Sharon
              HARRISON     08/29/91              Harrison, Ken
*NO GROUP
              DPTSM        09/05/91      X       Sales and Marketing
              DPTWH        09/18/91      X       Warehouse
```

Types of User Profile Displays: The Display User Profile (DSPUSRPRF) command provides several types of displays and listings:

- Some displays and listings are available only for individual profiles. Others can be printed for all profiles or a generic set of profiles. Consult online information for details about the available types.
- You can create an output file from some displays by specifying output(*OUTFILE). Use a query tool or program to produce customized reports from the output file. The topic "Analyzing User Profiles" on page 9-15 gives suggestions for reports.

Renaming a User Profile

The system does not provide a direct method for renaming a user profile. However, if a user changes names, you probably want to change that user's profile name. The following example shows how to create a new profile for a user with a new name and the same authorities. The old profile name is SMITHM. The new user profile name is JONESM:

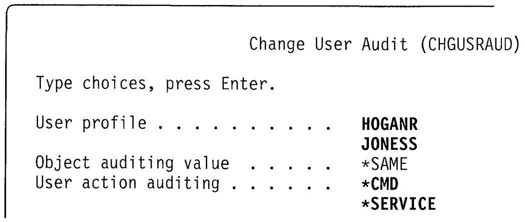
1. Copy the old profile (SMITHM) to a new profile (JONESM) using the copy option from the Work with User Enrollment display. The copy option from the Work with User Enrollment display copies the user's OfficeVision/400 enrollment. The copy option from the Work with User Profiles display does not copy OfficeVision/400 enrollment.
2. Give JONESM all the private authorities of SMITHM using the Grant User Authority (GRTUSRAUT) command:

   ```
   GRTUSRAUT JONESM REFUSER(SMITHM)
   ```
3. Change the ownership of any OfficeVision/400 objects owned by SMITHM using the Change Document Library Object Owner (CHGDLOOWN) command:

   ```
   CHGDLOOWN OWNER(SMITHM) NEWOWN(JONESM)
   ```
4. Transfer ownership of all other owned objects to JONESM and remove the SMITHM user profile, using option 4 (Remove) from the Work with User Enrollment display.

Working with User Auditing

Use the Change User Auditing (CHGUSRAUD) command to set the audit characteristics for users. To use this command, you must have *AUDIT special authority.



```
                    Change User Audit (CHGUSRAUD)

Type choices, press Enter.

User profile . . . . . . . . . .   HOGANR
                                   JONESS
Object auditing value  . . . . .   *SAME
User action auditing . . . . . .   *CMD
                                   *SERVICE
```

You can specify the auditing characteristics for more than one user at a time by listing user profile names.

The AUDLVL (user action auditing) parameter can have more than one value. The values you specify on this command replace the current AUDLVL values for the users. The values you specify are not added to the current AUDLVL values for the users.

You can use the Display User Profile (DSPUSRPRF) command to see audit characteristics for a user.

Working with Profiles in CL Programs

You may want to retrieve information about the user profile from within a CL program. You can use the Retrieve User Profile (RTVUSRPRF) command in your CL program. The command returns the requested attributes of the profile to variables you associate with the user profile field names. The descriptions of user profile fields in this chapter show the field lengths expected by the RTVUSRPRF command. In some cases, a decimal field can also have a value that is not numeric. For example, the maximum storage field (MAXSTG) is defined as a decimal field, but it can have a value of *NOMAX. Online information for the RVTUSRPRF command describes the values that are returned in a decimal field for values that are not numeric.

The sample program in "Using a Password Approval Program" on page 3-8 shows an example of using the RTVUSRPRF command.

You may also want to use the CRTUSRPRF or CHGUSRPRF command within a CL program. If you use variables for the parameters of these commands, define the variables as character fields to match the Create User Profile prompt display. The variable sizes do not have to match the field sizes.

You cannot retrieve a user's password, because the password is stored with one-way encryption. If you want the user to enter the password again before accessing critical information, you can use the Check Password (CHKPWD) command in your program. The system compares the password entered to the user's password and sends an escape message to your program if the password is not correct.

IBM-Supplied User Profiles

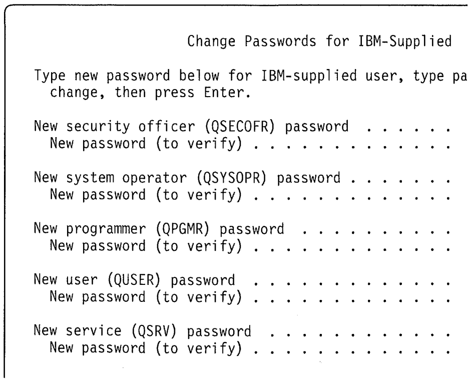
A number of user profiles are shipped with your system software. These IBM-supplied user profiles are used as object owners for various system functions. Some system functions also run under specific IBM-supplied user profiles.

Most IBM-supplied user profiles are shipped with a password of *NONE and are not intended for sign-on. A few IBM-supplied user profiles are designed as models of different types of users. These user profiles are shipped with passwords equal to the profile names. Because these passwords are the same for every AS/400 system that is shipped, you should change them as soon as your system is installed:

QPGMR	Programmer
QSECOFR	Security Officer
QSRV	Full Service Functions (display/alter)
QSRVBAS	Basic Service Functions
QSYSOPR	System Operator
QUSER	Work Station User

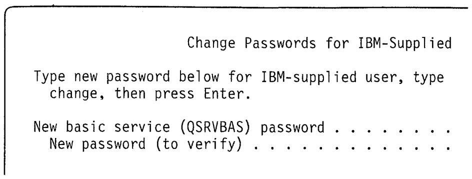
Appendix B contains a complete list of all the IBM-supplied user profiles and the field values for each profile.

Changing Passwords for IBM-Supplied User Profiles: You can change the passwords for IBM-supplied user profiles using the CHGUSRPRF command. You can also change these passwords using an option from the SETUP menu:



```
                    Change Passwords for IBM-Supplied

Type new password below for IBM-supplied user, type pa
  change, then press Enter.

New security officer (QSECOFR) password  . . . . . .
  New password (to verify) . . . . . . . . . . . . .

New system operator (QSYSOPR) password . . . . . . .
  New password (to verify) . . . . . . . . . . . . .

New programmer (QPGMR) password  . . . . . . . . . .
  New password (to verify) . . . . . . . . . . . . .

New user (QUSER) password  . . . . . . . . . . . . .
  New password (to verify) . . . . . . . . . . . . .

New service (QSRV) password  . . . . . . . . . . . .
  New password (to verify) . . . . . . . . . . . . .
```

Page down to change additional passwords:

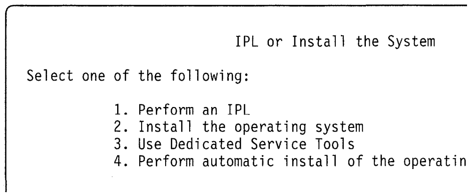


```
                    Change Passwords for IBM-Supplied

Type new password below for IBM-supplied user, type
  change, then press Enter.

New basic service (QSRVBAS) password . . . . . . . .
  New password (to verify) . . . . . . . . . . . . .
```

Changing Passwords for Dedicated Service Tools (DST) Users: DST is a set of tools for performing tests and service on your system outside the normal operating system. Three levels of DST are available, and a password is provided for each level. These passwords are the same for every AS/400 system that is shipped and should be changed to protect the security of your system.

You cannot change DST passwords using the CHGUSRPRF command. They can only be changed through the DST function. Use the following procedure:

1. With the keylock switch in the Manual position, start an attended Initial Program Load (IPL). When the system displays the IPL or Install the System menu, select option 3 (Use Dedicated Service Tools):



```
                           IPL or Install the System

Select one of the following:

          1. Perform an IPL
          2. Install the operating system
          3. Use Dedicated Service Tools
          4. Perform automatic install of the operatin
```

2. Type the DST security capability password on the Dedicated Service Tools (DST) Sign On display. When your system is shipped, this password is QSECOFR.
3. Select menu options in this sequence:

Menu or display name	*Select this option:*
Use Dedicated Service Tools (DST) menu	Option 5 (Work with DST environment)
Work with DST Environment menu	Option 9 (Change DST passwords)
Change DST Password menu	Option 1 (Change the DST basic capability password)
Change DST Password menu	Option 2 (Change the DST full capability password)
Change DST Password menu	Option 3 (Change the DST security capability password)

   Note: In the *Current password* field for the basic capability or full capability profiles, you can type either the current password for that profile or the DST security password. If you forget the basic or full capability password, you can use the security capability password to assign a new one.

4. To leave DST, press F3 (Exit) until you return to the IPL or Install the System menu. Continue with a normal IPL.

Warning:

- Write down the passwords you assign and keep them in a safe place. If you lose or forget both the QSECOFR and the DST security capability passwords, you may need to install your operating system again to recover them. Contact your service provider for assistance. The topic "Recovering a Lost DST or QSECOFR Password" on page 4-24 tells how to recover one of these passwords if you know the other password.
- You must provide the DST basic capability password whenever your system needs service. Your system cannot be serviced without this password.
- Change the DST passwords on your system after service personnel have finished using them.

Recovering a Lost DST or QSECOFR Password: If you know either the QSECOFR password or the DST security capability password, you can reset the other one. You can also change the DST full capability password and the DST basic capability password if you know the DST security capability password.

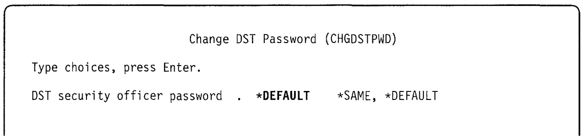
Resetting the QSECOFR Password: You can use the DST security capability password to reset the QSECOFR password to its initial value (QSECOFR):

1. The topic "Changing Passwords for Dedicated Service Tools (DST) Users" on page 4-23 describes how to reach the Change DST Password menu.
2. Select option 4 (Reset system default password).
3. You receive a message confirming that the QSECOFR password has been reset to its default value (QSECOFR).
4. Continue pressing F3 (Exit) to return to the IPL or Install the System menu. Select option 1 (perform an IPL).
5. When the IPL has completed, return the keylock to the Auto position.
6. Sign on as QSECOFR. Use the CHGPWD command to change the QSECOFR password to a new value. Write down the new value and store it in a safe place.

   Warning: Do not leave the QSECOFR password set to the default. This poses a security exposure, because this is the value shipped with every system and is commonly known.

Resetting the DST Security Capability Password: If you know the password for the QSECOFR profile, you can reset the DST security capability password to the initial setting (QSECOFR):

1. The system should be in normal operating mode (not DST). Sign on at any workstation using the QSECOFR profile.
2. On a command line, type `CHGDSTPWD` (Change DST Password). You see the Change DST Password (CHGDSTPWD) display:



```
                         Change DST Password (CHGDSTPWD)

 Type choices, press Enter.

 DST security officer password  .  *DEFAULT    *SAME, *DEFAULT
```

3. Type `*DEFAULT` and press the Enter key. The DST security capability password is set to QSECOFR.
4. Perform an attended IPL and use DST to change the DST security capability password to another value. (See the topic "Changing Passwords for Dedicated Service Tools (DST) Users" on page 4-23 for detailed instructions.)
5. Write down the new value and store it in a safe place.

   Warning: Do not leave the DST security capability password set to the default. This poses a security exposure, because this is the value shipped with every system and is commonly known.

Chapter 5. Resource Security

Resource security defines which users are allowed to use objects on the system and what operations they are allowed to perform on those objects.

This chapter describes each of the components of resource security and how they all work together to protect information on your system. It also explains how to use CL commands and displays to set up resource security on your system.

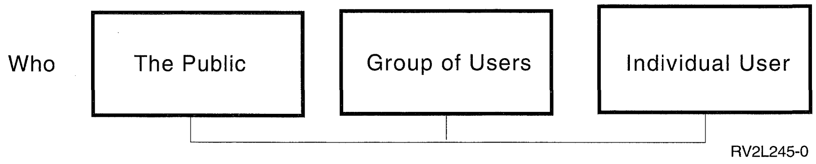
Chapter 7 discusses techniques for designing resource security, including how it affects both application design and system performance.

The topic "How the System Checks Authority" on page 5-9 provides detailed flowcharts and notes about how the system checks authority. You may find it useful to consult this information as you read the explanations that follow.

This chapter does not discuss the methods available for securing OfficeVision/400 documents and folders. Consult the *Office Services Concepts and Programmer's Guide* for information about OfficeVision/400 security.

Defining Who Can Access Information

You can give authority to individual users, groups of users, and the public.



You define who can use an object in several ways:

Public Authority: **The public** consists of anyone who is authorized to sign on to your system. Public authority is defined for every object on the system, although the public authority for an object may be *EXCLUDE. Public authority to an object is used if no other specific authority is found for the object.

Private Authority: You can define specific authority to use (or not use) an object. You can grant authority to an individual user profile or to a group profile. An object has **private authority** if any authority other than public authority and object ownership is defined for the object.

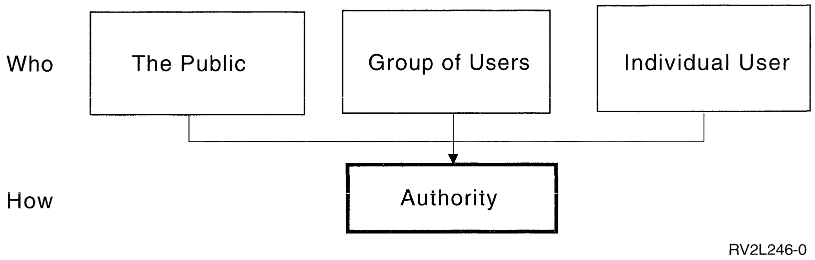
User Authority: Individual user profiles may be given authority to use objects on the system. This is one type of private authority.

Group Authority: Group profiles may be given authority to use objects on the system. A member of the group gets the group's authority unless an authority is specifically defined for that user. Group authority is also considered private authority.

Object Ownership: Every object on the system has an owner. The owner has *ALL authority to the object by default. However, the owner's authority to the object can be changed or removed. The owner's authority to the object is not considered private authority.

Defining How Information Can Be Accessed

Authority means the type of access allowed to an object. Different operations require different types of authority.



Authority to an object is divided into two categories: 1) **Object Authority** defines what operations can be performed on the object as a whole. 2) **Data Authority** defines what operations can be performed on the contents of the object.

Table 5-1 describes the types of authority available:

Table 5-1. Description of Authority Types

Authority Name	Descriptive Name	Functions Allowed
Object Authorities:		
*OBJOPR	Object Operational	Look at the description of an object. Use the object as determined by the user's data authorities.
*OBJMGT	Object Management	Specify the security for the object. Move or rename the object. Add members to database files.
*OBJEXIST	Object Existence	Delete the object. Free storage of the object. Perform save and restore operations for the object [1]. Transfer ownership of the object.
*AUTLMGT	Authorization List Management	Add and remove users and their authorities from the authorization list [2].
Data Authorities:		
*READ	Read	Display the contents of the object, such as viewing records in a file. Run a program. Access the objects in a library.
*ADD	Add	Add entries to an object such as adding jobs to a job queue or adding records to a file.
*UPD	Update	Change the entries in an object, such as changing records in a file.
*DLT	Delete	Remove entries from an object, such as removing messages from a message queue or deleting records from a file.

[1] If a user has save system (*SAVSYS) special authority, object existence authority is not required to perform save and restore operations on the object.

[2] See the topic "Authorization List Management" on page 5-4 for more information.

Commonly Used Authorities

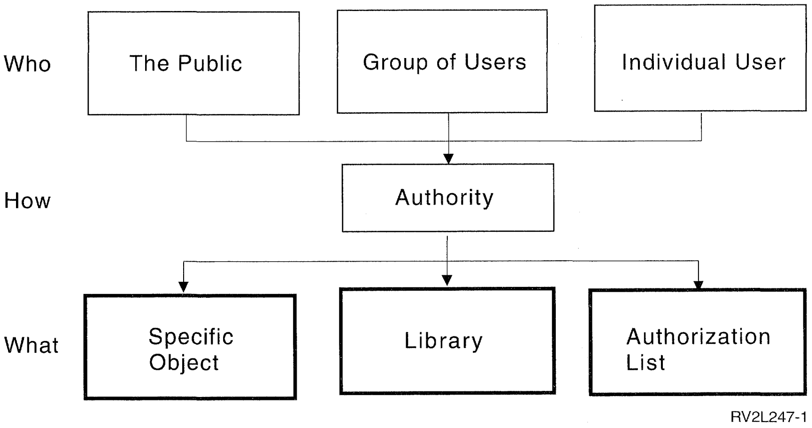
Certain sets of object and data authorities are commonly required to perform operations on objects. You can specify these system-defined sets of authority (*ALL, *CHANGE, *USE) instead of individually defining the authorities needed for an object. *EXCLUDE authority is different than having no authority. *EXCLUDE authority specifically denies access to the object. Having no authority means you use the public authority defined for the object.

Table 5-2. System-Defined Authority

Authority	*ALL	*CHANGE	*USE	*EXCLUDE
Object Authorities				
*OBJOPR	X	X	X	
*OBJMGT	X			
*OBJEXIST	X			
Data Authorities				
*READ	X	X	X	
*ADD	X	X		
*UPD	X	X		
*DLT	X	X		

Defining What Information Can Be Accessed

You can define resource security for individual objects on the system. You can also define security for groups of objects using either library security or an authorization list:



Library Security

Most objects on the system reside in libraries. To access an object, you need authority both to the object itself and the library in which the object resides. For most operations, including deleting an object, *USE authority to the object library is sufficient (in addition to the authority required for the object). Creating a new object requires *ADD authority to the object library. Appendix D shows what authority is required by CL commands for objects and the object libraries.

Using library security is one technique for protecting information while maintaining a simple security scheme. For example, to secure confidential information for a set of applications, you could do the following:

- Use a library to store all confidential files for a particular group of applications.
- Make public authority for all the objects in the library sufficient for the application needs (*CHANGE or *ALL).
- Restrict public authority to the library itself (*EXCLUDE).
- Give selected groups or individuals authority to the library (*USE, or *ADD if the applications require it).

Although library security is a simple, effective method for protecting information, it may not be adequate for data with high security requirements. In some situations, knowledgeable users who are authorized to commands and programming languages may be able to circumvent library security. Highly sensitive objects should be secured individually or with an authorization list, rather than relying on library security.

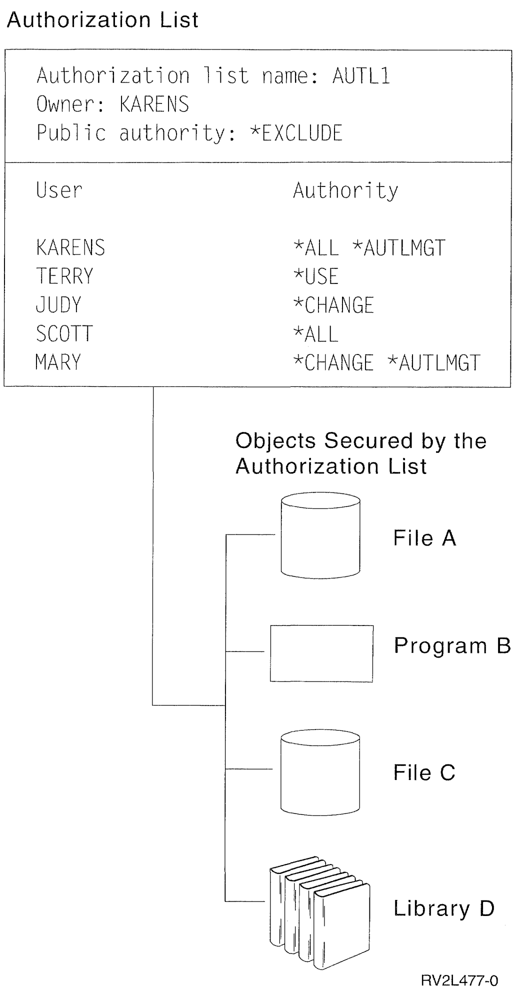
Library Security and Library Lists: When a library is added to a user's library list, the authority the user has to the library is stored with the library list information. The user's authority to the library remains for the entire job, even if the user's authority to the library is revoked while the job is active.

When access is requested to an object and *LIBL is specified for the object, the library list information is used to check authority for the library. If a qualified name is specified, the authority for the library is specifically checked, even if the library is included in the user's library list.

Warning: If a user is running under adopted authority when a library is added to the library list, the adopted authority remains with the library list entry even when the user is no longer running under adopted authority. This represents a potential security exposure. Any entries added to a user's library list by a program running under adopted authority should be removed before the adopted authority program ends.

Authorization List Security

You can group objects with similar security requirements using an authorization list. An authorization list conceptually contains a list of users and the authority that the users have to the objects secured by the list. Each user can have a different authority to the set of objects the list secures:



```
Authorization List

Authorization list name: AUTL1
Owner: KARENS
Public authority: *EXCLUDE

User                Authority

KARENS              *ALL *AUTLMGT
TERRY               *USE
JUDY                *CHANGE
SCOTT               *ALL
MARY                *CHANGE *AUTLMGT
```

Figure 5-1. Example of an Authorization List (Conceptual Representation)

You can also use an authorization list to define public authority for the objects on the list. If the public authority for an object is set to *AUTL, the object gets its public authority from its authorization list.

The authorization list object is used as a management tool by the system. It actually contains a list of all objects which are secured by the authorization list. This information is used to build displays for viewing or editing the authorization list objects.

You cannot use an authorization list to secure a user profile or another authorization list. Only one authorization list can be specified for an object.

Only the owner of the object, a user with all object (*ALLOBJ) special authority, or a user with all (*ALL) authority to the object, can add or remove the authorization list for an object.

Objects in the system library (QSYS) can be secured with an authorization list. However, the name of the authorization list that secures an object is stored with the object. In some cases, when you install a new release of the operating system, all the objects in the QSYS library are replaced. The association between the objects and your authorization list would be lost.

See the topic "Planning Authorization Lists" on page 7-9 for examples of how to use authorization lists.

Authorization List Management: You can grant a special operational authority called Authorization List Management (*AUTLMGT) for authorization lists. Users with *AUTLMGT authority are allowed to add and remove users on the list and change those users' authorities to the list. Changes are made to user profiles to affect the authority for each user. *AUTLMGT authority does not by itself give authority to secure new objects with the list or to remove objects from the list.

A user with *AUTLMGT authority can give only the same or less authority to others. For example, assume USERA has *CHANGE and *AUTLMGT authority to authorization list CPLIST1. USERA can add USERB to CPLIST1 and give USERB *CHANGE authority or less. USERA cannot give USERB *ALL authority to CPLIST1, because USERA does not have *ALL authority.

A user with *AUTLMGT authority can remove a user from the list only if the *AUTLMGT user has equal or greater authority to the list than the user profile name being removed. If USERC has *ALL authority to CPLIST1, then USERA cannot remove USERC from the list, because USERA has only *CHANGE and *AUTLMGT.

Using Authorization Lists to Secure IBM-Supplied Objects: You may choose to use an authorization list to secure IBM-supplied objects. For example, you may want to restrict the use of a group of commands to a few users.

Objects in IBM-supplied libraries, other than the QUSRSYS and QGPL libraries, are replaced whenever you install a new release of the operating system. Therefore, the link between objects in IBM-supplied libraries and authorization lists is lost. After you install a new release, use the EDTOBJAUT or GRTOBJAUT command to establish the link between the IBM-supplied object and the authorization list again.

Authority for New Objects in a Library

Every library has a parameter called CRTAUT (create authority). This parameter determines the default public authority for any new object that is created in that library. When you create an object, the AUT parameter on the create command determines the public authority for the object. If the AUT value on the create command is *LIBCRTAUT, which is the default, the public authority for the object is set to the CRTAUT value for the library.

For example, assume library CUSTLIB has a CRTAUT value of *USE. Both of the commands below create a data area called DTA1 with public authority *USE:

- Specifying the AUT parameter:

  ```
  CRTDTAARA DTAARA(CUSTLIB/DTA1) +
     TYPE(*CHAR) AUT(*LIBCRTAUT)
  ```

- Allowing the AUT parameter to default. *LIBCRTAUT is the default:

  ```
  CRTDTAARA DTAARA(CUSTLIB/DTA1) +
     TYPE(*CHAR)
  ```

The default CRTAUT value for a library is *SYSVAL. Any new objects created in the library using AUT(*LIBCRTAUT) have public authority set to the value of the QCRTAUT system value. The QCRTAUT system value is shipped as *CHANGE. For example, assume the ITEMLIB library has a CRTAUT value of *SYSVAL. This command creates the DTA2 data area with public authority of change:

```
CRTDTAARA DTAARA(ITEMLIB/DTA2) +
   TYPE(*CHAR) AUT(*LIBCRTAUT)
```

Warning: Several IBM-supplied libraries, including QSYS, have a CRTAUT value of *SYSVAL. If you change QCRTAUT to something other than *CHANGE, you may encounter problems. For example, devices are created in the QSYS library. The default when creating devices is AUT(*LIBCRTAUT). The CRTAUT value for the QSYS library is *SYSVAL. If QCRTAUT is set to *USE or *EXCLUDE, public authority is not sufficient to allow sign-on at new devices.

The CRTAUT value for a library can also be set to an authorization list name. Any new object created in the library with AUT(*LIBCRTAUT) is secured by the authorization list. The public authority for the object is set to *AUTL.

The CRTAUT value of the library is not used during a move (MOVOBJ), create duplicate (CRTDUPOBJ), or restore of an object into the library. The public authority of the existing object is used.

If the REPLACE (*YES) parameter is used on the create command, then the authority of the existing object is used instead of the CRTAUT value of the library.

Create Authority (CRTAUT) Risks

If your applications use default authority for new objects created during application processing, you should control who has authority to change the library descriptions. Changing the CRTAUT authority for an application library could allow unauthorized access to new objects created in the library.

Object Ownership

Each object is assigned an owner when it is created. The owner is either the user who creates the object or the group profile if the member user profile has specified that the group profile should be the owner of the object. When the object is created, the owner is given all the object and data authorities to the object.

The owner of an object always has all the authority for the object unless any or all authority is removed specifically. As an object owner, you may choose to remove some specific authority as a precautionary measure. For example, if a file exists that contains critical information, you may remove your object existence authority to prevent yourself from accidentally deleting the file. However, as object owner, you can grant any object authority to yourself at any time.

Ownership of an object can be transferred from one user to another. When changing an object's owner, you have the option to keep or revoke the former owner's authority. A user with *ALLOBJ authority can transfer ownership, as can any user who has the following:

- Object existence authority for the object (except for an authorization list)
- Ownership of the object, if the object is an authorization list
- Add authority for the new owner's user profile
- Delete authority for the present owner's user profile

You cannot delete a profile that owns objects. Ownership of objects must be transferred to a new owner or the objects must be deleted before the profile can be deleted. The Delete User Profile (DLTUSRPRF) command allows you to handle owned objects when you delete the profile.

Object ownership is used as a management tool by the system. The owner profile for an object contains a list of all users who have private authority to the object. This information is used to build displays for editing or viewing object authority.

Profiles that own many objects with many private authorities can become very large. The size of an owner profile affects performance when displaying and working with the authority to owned objects, and when saving or restoring profiles. Take this into consideration when planning object ownership and private authorities.

The owner of an object also needs sufficient storage for the object. See "Maximum Storage" on page 4-10 for more information.

Group Ownership of Objects

When an object is created, the system looks at the profile of the user creating the object to determine object ownership. If the user is a member of a group profile, the OWNER field in the user profile specifies whether the user or the group should own the new object.

If the group owns the object (OWNER is *GRPPRF), the user creating the object is not automatically given any specific authority to the object. The user gets authority to the object through the group. If the user owns the object (OWNER is *USRPRF), the group's authority to the object is determined by the GRPAUT field in the user profile. The group authority becomes a private authority to the object. If the user who owns the object changes to a different user group, the original group profile still retains authority to any objects created.

Even if the *Owner* field in a user profile is *GRPPRF, the user must still have sufficient storage to hold a new object while it is being created. After it is created, ownership is transferred to the group profile. The MAXSTG parameter in the user profile determines how much auxiliary storage a user is allowed.

Evaluate the objects a user might create, such as query programs, when choosing between group and individual user ownership:

- If the user moves to a different department and a different user group, should the user still own the objects?
- Is it important to know who creates objects? The object authority displays show the object owner, not the user who created the object.

  Note: The Display Object Description display shows the object creator.

  If the audit journal function is active, a Create Object (CO) entry is written to the QAUDJRN audit journal at the time an object is created. This entry identifies the creating user profile. The entry is written only if the QAUDLVL system value specifies *CREATE.

Default Owner (QDFTOWN) User Profile

The Default Owner (QDFTOWN) user profile is an IBM-supplied user profile that is used when an object has no owner or when object ownership might pose a security exposure. Following are situations that cause ownership of an object to be assigned to the QDFTOWN profile:

- If an owning profile becomes damaged and is deleted, its objects no longer have an owner. Using the Reclaim Storage I2.QDFTOWN (default owner) profile (RCLSTG) command assigns ownership of these objects to the default owner (QDFTOWN) user profile.
- If an object is restored and the owner profile does not exist.
- If a program that needs to be created again is restored, but the program creation is not successful. See the topic "Validation of Programs Being Restored" on page 2-6 for more information about which conditions cause ownership to be assigned to QDFTOWN.
- If the maximum storage limit is exceeded for the user profile that owns an authority holder that has the same name as a file being moved, renamed, or whose library is being renamed.

The system supplies the QDFTOWN user profile because all objects must have an owner. When the system is shipped, only a user with *ALLOBJ special authority can display and access this user profile and transfer ownership of objects associated with the QDFTOWN user profile. You can grant other users authority to the QDFTOWN profile.

Objects That Adopt the Owner's Authority

Sometimes a user needs different authorities to an object or an application, depending on the situation. For example, a user may be allowed to change the information in a customer file when using application programs providing that function. However, the same user should be allowed to view, but not change, customer information when using a decision support tool, such as SQL.

A solution to this situation is 1) give the user *USE authority to customer information to allow querying the files and 2) use adopted authority in the customer maintenance programs to allow the user to change the files.

When an object uses the owner's authority, this is called **adopted authority**. Objects of type *PGM, *SRVPGM, and *SQLPKG can adopt authority.

When you create a program, you specify a user profile (USRPRF) parameter on the CRTxxxPGM command. This parameter determines whether the program uses the authority of the owner of the program in addition to the authority of the user running the program.

The *Systems Application Architecture* Structured Query Language/400 Programmer's Guide* describes security considerations and adopted authority when using SQL packages

The following applies to adopted authority:

- Adopted authority is added to any other authority found for the user.
- Adopted authority is checked only if the authority that the user, the user's group, or the public has to an object is not adequate for the requested operation.
- The special authorities (such as *ALLOBJ) in the owner's profile are used.
- If the owner profile is a member of a group profile, the group's authority is *not* used for adopted authority.

- Public authority is *not* used for adopted authority. For example, USER1 runs the program LSTCUST, which requires *USE authority to the CUSTMST file:
  - Public authority to the CUSTMST file is *USE.
  - USER1's authority is *EXCLUDE.
  - USER2 owns the LSTCUST program, which adopts owner authority.
  - USER2 does not own the CUSTMST file and has no private authority to it.
  - Although public authority is sufficient to give USER2 access to the CUSTMST file, USER1 does not get access. Only owner authority and private authority are used for adopted authority.
- Adopted authority is active as long as the program using adopted authority remains in the program stack. For example, assume PGMA uses adopted authority:
  - If PGMA starts PGMB using the CALL command, these are the program stacks before and after the CALL command:

Program Stack before CALL Command:	**Program Stack after CALL Command:**
QCMD ⋮ PGMA	QCMD ⋮ PGMA PGMB

Figure 5-2. Adopted Authority and the CALL Command

    Because PGMA remains in the program stack after PGMB is called, PGMB uses the adopted authority of PGMA. (The use adopted authority (USEADPAUT) parameter can override this. See "Programs That Ignore Adopted Authority" on page 5-8 for more information about the USEADPAUT parameter.)

  - If PGMA starts PGMB using the Transfer Control (TFRCTL) command, the program stacks look like this:

Program Stack before TFRCTL Command:	**Program Stack after TFRCTL Command:**
QCMD ⋮ PGMA	QCMD ⋮ PGMB

Figure 5-3. Adopted Authority and the TFRCTL Command

    PGMB does not use the adopted authority of PGMA, because PGMA is no longer in the program stack.

- If the program running under adopted authority is interrupted, the use of adopted authority is suspended. The following functions do not use adopted authority:
  - System request
  - Attention key (If a Transfer to Group Job (TFRGRPJOB) command is running, adopted authority is not passed to the group job.)
  - Break-message-handling program
  - Debug functions

  Note: Adopted authority is immediately interrupted by the attention key or a group job request. The user must have authority to the attention-key-handling program or the group job initial program, or the attempt fails.

  For example, USERA runs the program PGM1, which adopts the authority of USERB. PGM1 uses the SETATNPGM command and specifies PGM2. USERB has *USE authority to PGM2. USERA has *EXCLUDE authority to PGM2. The SETATNPGM function is successful because it is run using adopted authority. USERA receives an authority error when attempting to use the attention key because USERB's authority is no longer active.

- If a program that uses adopted authority submits a job, that submitted job does not have the adopted authority of the submitting program.
- The program adopt function is not used when a change occurs to the job queue or output queue parameters on the Change Job (CHGJOB) command. The user profile must have authority to the queue to change these parameters.
- Any objects created, including spooled files that may contain confidential data, are owned by the user of the program or by the user's group profile, not by the owner of the program.
- Adopted authority can be specified on either the command that creates the program (CRTxxxPGM) or on the Change Program (CHGPGM) command.
- If a program is created using REPLACE(*YES) on the CRTxxxPGM command, the new copy of the program has the same USRPRF, USEADPAUT, and AUT values as the replaced program. The USRPRF and AUT parameters specified on the CRTxxxPGM parameter are ignored.
- Only a user who owns the program or has *ALLOBJ and *SECADM special authorities can change the value of the USRPRF parameter.
- You must be signed on as a user with *ALLOBJ and *SECADM special authorities to transfer ownership of an object that adopts authority.
- If someone other than the program's owner or a user with *ALLOBJ and *SECADM special authorities restores a program that adopts authority, all private and public authorities to the program are revoked to prevent a possible security exposure.

The Display Program (DSPPGM) and Display Service Program (DSPSRVPGM) commands show whether a program adopts authority (*User profile* prompt) and whether it uses adopted authority from previous programs in the program stack (*Use adopted authority* prompt). The Display

Program Adopt (DSPPGMADP) command shows all the objects that adopt the authority of a specific user profile.

"Flowchart 6: How Adopted Authority Is Checked" on page 5-15 provides more information about adopted authority. The topic "Using Adopted Authority in Menu Design" on page 7-4 shows an example of how to use adopted authority in an application.

Adopted Authority and Bound Programs: If a bound program uses adopted authority, the adopted authority does not take effect until the bound program and any service programs are activated. To activate a bound program successfully, the user must have *USE authority to the bound program and to all related service programs. Adopted authority is not used to acquire *USE authority to the service programs.

Adopted Authority Risks and Recommendations

Allowing a program to run using adopted authority is an intentional release of control. You permit the user to have authority to objects, and possibly special authority, which the user would not normally have. Adopted authority provides an important tool for meeting diverse authority requirements, but it should be used with care:

- Adopt the minimum authority required to meet the application requirements. Adopting the authority of an application owner is preferable to adopting the authority of QSECOFR or a user with *ALLOBJ authority.
- Carefully monitor the function provided by programs that adopt authority. Make sure these programs do not provide a means for the user to access objects outside the control of the program, such as command entry capability.
- Control which users are permitted to call programs that adopt authority. Use menu interfaces and library security to prevent these programs from being called without sufficient control.

Programs That Ignore Adopted Authority

You may not want some programs to use the adopted authority of previous programs in the program stack. For example, if you use an initial menu program that adopts owner authority, you may not want some of the programs called from the menu program to use that authority.

The use adopted authority (USEADPAUT) parameter of a program determines whether the system uses the adopted authority of previous programs in the stack when checking authority for objects.

When you create a program, the default is to use adopted authority. You must use the Change Program (CHGPGM) or Change Service Program (CHGSRVPGM) command to set the USEADPAUT parameter to *NO. If a program is created using REPLACE(*YES) on the CRTxxxPGM command, the new copy of the program has the same USRPRF, USEADPAUT, and AUT values as the replaced program.

The topic "Ignoring Adopted Authority" on page 7-6 shows an example of how to use this parameter in menu design.

Authority Holders

An authority holder is a tool for keeping the authorities for a program-described database file that does not currently exist on the system. Its primary use is for System/36 environment applications, which often delete program-described files and create them again.

An authority holder can be created for a file that already exists or for a file that does not exist, using the Create Authority Holder (CRTAUTHLR) command. The following applies to authority holders:

- The authority holder is associated with a specific file and library. It has the same name as the file.
- Authority holders can be used only for program-described database files.
- Once the authority holder is created, you add private authorities for it like a file. Use the commands to grant, revoke, and display object authorities, and specify object type *FILE. On the object authority displays, the authority holder is indistinguishable from the file itself. The displays do not indicate whether the file exists nor do they show that the file has an authority holder.
- If a file is associated with an authority holder, the authorities defined for the authority holder are used during authority checking. Any private authorities defined for the file are ignored.
- Use the Display Authority Holder (DSPAUTHLR) command to display or print all the authority holders on the system. You can also use it to create an output file (Outfile) for processing.
- If you create an authority holder for a file that exists:
  - The user creating the authority holder must have *ALL authority to the file.
  - The owner of the file becomes the owner of the authority holder regardless of the user creating the authority holder.
  - The public authority for the authority holder comes from the file. The public authority (AUT) parameter on the CRTAUTHLR command is ignored.
  - The existing file's authority is copied to the authority holder.
- If you create a file and an authority holder for that file already exists:
  - The user creating the file must have *ALL authority to the authority holder.

  - The owner of the authority holder becomes the owner of the file regardless of the user creating the file.
  - The public authority for the file comes from the authority holder. The public authority (AUT) parameter on the CRTPF or CRTLF command is ignored.
  - The authority holder is linked to the file. The authority specified for the authority holder is used to secure the file.
  - If the file is a logical file, no data authorities are used because data authorities are not valid for logical files.

- If an authority holder is deleted, the authority information is transferred to the file itself.
- If a file is renamed and the new file name matches an existing authority holder, the authority and ownership of the file are changed to match the authority holder. The user renaming the file needs *ALL authority to the authority holder.
- If a file is moved to a different library and an authority holder exists for that file name and the target library, the authority and ownership of the file are changed to match the authority holder. The user moving the file must have *ALL authority to the authority holder.
- Ownership of the authority holder and the file always match. If you change the ownership of the file, ownership of the authority holder also changes.
- When a file is restored, if an authority holder exists for that file name and the library to which it is being restored, it is linked to the authority holder.
- Authority holders cannot be created for files in these libraries: QSYS, QRCL, QRECOVERY, QSPL, and QTEMP.

Authority Holders and System/36 Migration

The System/36 Migration Aid creates an authority holder for every file that is migrated. It also creates an authority holder for entries in the System/36 resource security file if no corresponding file exists on the System/36.

You need authority holders only for files that are deleted and re-created by your applications. Use the Delete Authority Holder (DLTAUTHLR) command to delete any authority holders that you do not need.

Authority Holder Risks

An authority holder provides the capability of defining authority for a file before that file exists. Under certain circumstances, this could allow an unauthorized user to gain access to information. If a user knew that an application would create, move, or rename a file, the user could create an authority holder for the new file. The user would thus gain access to the file.

To limit this exposure, the CRTAUTHLR command is shipped with public authority *EXCLUDE. Only users with *ALLOBJ authority can use the command, unless you grant authority to others.

How the System Checks Authority

When a user attempts to perform an operation on an object, the system verifies that the user has adequate authority for the operation. The system first checks authority to the object library. If the authority to the library is adequate, the system checks authority to the object itself. In the case of database files, authority checking is done at the time the file is opened, not when each individual operation to the file is performed.

During the authority-checking process, when any authority is found (even if it is not adequate for the requested operation) authority checking stops and access is granted or denied. The adopted authority function is the exception to this rule. Adopted authority can override any specific (and inadequate) authority found. See the topic "Objects That Adopt the Owner's Authority" on page 5-6 for more information about adopted authority.

The system verifies a user's authority to an object in the following order:

1. User's *ALLOBJ special authority
2. User's specific authority to the object
3. User's authority on the authorization list securing the object
4. Group's *ALLOBJ special authority
5. Group's authority to the object
6. Group's authority on the authorization list securing the object
7. Public authority specified for the object or for the authorization list securing the object
8. Program owner's authority, if adopted authority is used

Authority Checking Flowcharts

Following are charts, descriptions, and examples of how authority is checked. Use them to answer specific questions about whether a particular authority scheme will work or diagnose problems with your authority definitions. The charts also highlight the types of authority that cause the greatest performance impact.

The process of checking authority is divided into a primary flowchart and several smaller flowcharts showing specific parts of the process. Depending on the combination of authorities for an object, the steps in some flowcharts may be repeated several times.

In the flowcharts, a box with a double line at the top indicates a process that is described by another flowchart. The numbers at the upper left of figures on the flowcharts correspond to the step numbers in the descriptions of the flowcharts.

In Flowchart 4 on page 5-14, the box representing the search of a user's private authorities (step 8) is highlighted. Repeating this step is likely to cause performance problems in the authority checking process. The following fields are used in these flowcharts to define what is currently being checked:

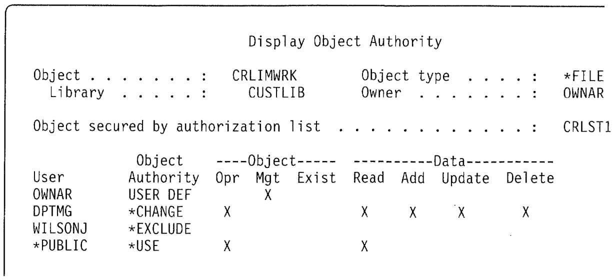
Object to check The name, library, and type of the object for which authority is being checked.

Profile to check The name of the user profile whose authority to the object is being checked.

Program to check The name of the program which is being tested for adopted authority.

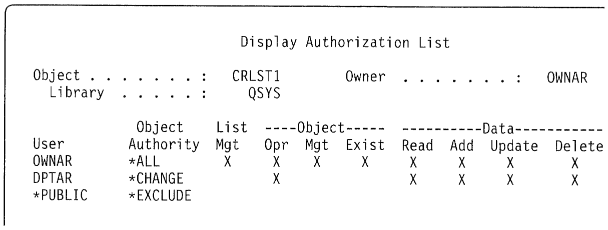
Result The results of the current process. Possible values are: *No authority found*, *Sufficient* (authority), or *Insufficient* (authority).

The explanations of the flowcharts use the CRLIMWRK file as an example. Figure 5-4 shows the authority for the CRLIMWRK file. Figure 5-5 shows the authority for the CRLST1 authorization list. In the example, group profiles start with the characters DPT. Owner profiles start with the characters OWN. This example shows most of the possibilities for authority checking. It also demonstrates how using too many authority options for an object can result in poor performance.



```
                         Display Object Authority

Object . . . . . . . :   CRLIMWRK        Object type  . . . . :   *FILE
  Library  . . . . . :     CUSTLIB       Owner  . . . . . . . :   OWNAR

Object secured by authorization list  . . . . . . . . . . . . :   CRLST1

             Object    ----Object-----  ----------Data-----------
User         Authority Opr  Mgt  Exist  Read  Add  Update  Delete
OWNAR        USER DEF        X
DPTMG        *CHANGE    X                X     X     X       X
WILSONJ      *EXCLUDE
*PUBLIC      *USE       X                X
```

Figure 5-4. Authority for CRLIMWRK File



```
                         Display Authorization List

Object . . . . . . . :   CRLST1          Owner  . . . . . . . :   OWNAR
  Library  . . . . . :     QSYS

             Object    List  ----Object-----  ----------Data-----------
User         Authority Mgt   Opr  Mgt  Exist  Read  Add  Update  Delete
OWNAR        *ALL       X     X    X     X     X     X     X       X
DPTAR        *CHANGE          X                X     X     X       X
*PUBLIC      *EXCLUDE
```

Figure 5-5. Authority for the CRLST1 Authorization List

The "Summary of the Flowchart Example" on page 5-20 summarizes all the steps used in this example. "Examples of Authority Checking" on page 5-20 shows additional examples of authority checking and highlights their performance characteristics.

Flowchart 1: Main Authority Checking Process: The steps in Flowchart 1 show the main process the system follows in checking authority for an object.

In the example, a user named WAGNERB, who is a member of group profile DPTAR and has no special authorities, wants to run a program that clears a member in the CRLIMWRK file. The CLRPFM (Clear Physical File Member) command requires operational, management, and delete authority to the file. (See page D-22.)

Step	Description
1	For an interactive job, the original profile is the user who signed on. For a batch job, the original profile is the user whose profile the job is running under. In the example, the profile to check is WAGNERB.
2	User authority to the object is checked. See Flowchart 2 on page 5-12.
3	If the result is *Sufficient*, the user is authorized to the object. If the result is *Insufficient* (authority was found but it was not sufficient for the operation requested), additional authority checking for WAGNERB is skipped and adopted authority is checked (step 8). If the result is *No authority found*, checking continues.
4	The profile to check is tested for membership in a group profile.
5	If the profile is a member of a group, the profile to test field is set to the group profile name. Checking is repeated for the group profile, starting at step 2. In the example, the profile to check is now DPTAR.
6	If the profile is not a group member, public authority is checked. See Flowchart 5 on page 5-15.
7	If public authority is sufficient, access is authorized. If public authority is not sufficient, checking continues.
8	Flowchart 6 on page 5-16 shows the process for checking adopted authority.
9	When the adopted authority check is completed, the user is either authorized or not authorized to the object. If the user is not authorized, one of more of the following happens: • A message is sent to the user or program. • The program fails. • An AF (authority failure) entry is written to the audit journal, if the auditing function is active.

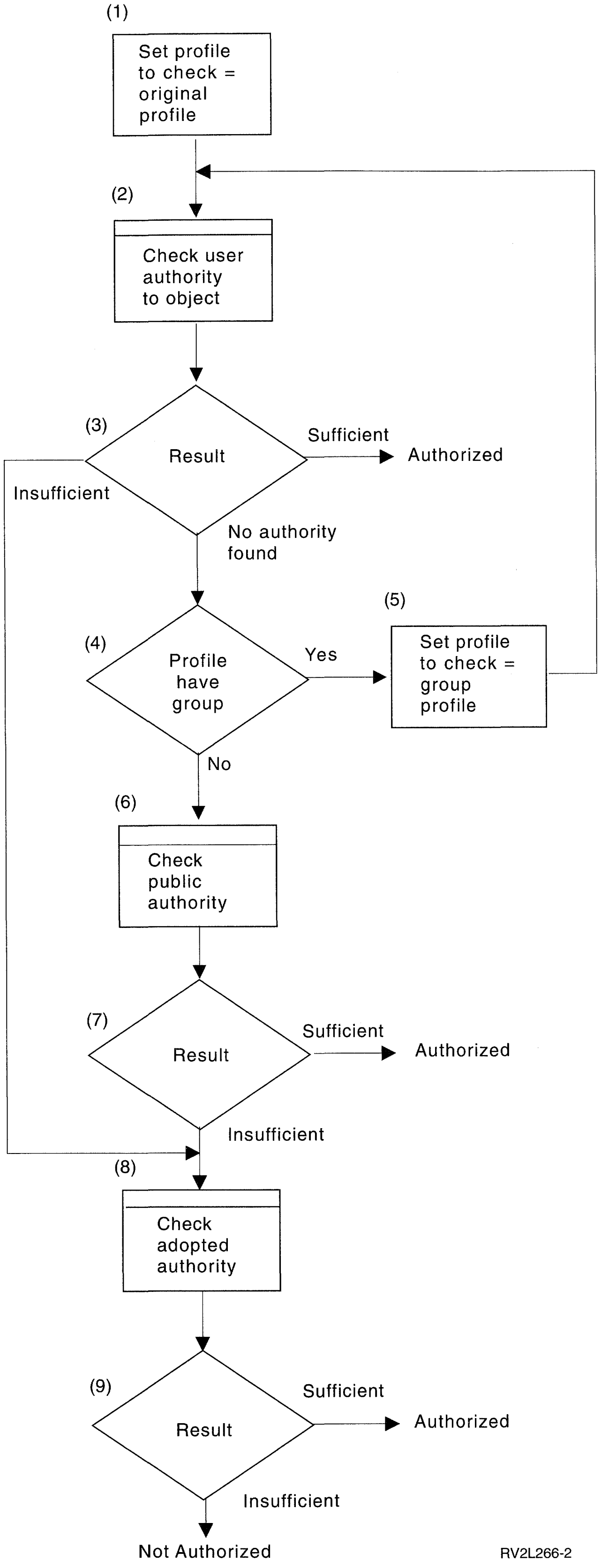


Figure 5-6. Flowchart 1: Main Authority Checking Process

Flowchart 2: How User Authority to an Object Is Checked: The steps in Flowchart 2 may be performed for both the individual user profile and the user's group profile. In some cases, the steps in this chart may also be repeated one or more times as part of the adopted authority check (see Flowchart 6 on page 5-16).

Step Description

1 The profile is checked for *ALLOBJ special authority. In the example, neither WAGNERB nor DPTAR (group profile for WAGNERB) has *ALLOBJ special authority.

2 If the profile has *ALLOBJ special authority, the result is set to *Sufficient* and the remaining steps in the flowchart are skipped.

3 The object to check field is set to the original object being requested. In the example, this is the CRLIMWRK file in the CUSTLIB library.

4 A test is performed to see if the profile owns the object and has adequate authority. See Flowchart 3 on page 5-13.

5 If the profile owns the object and has authority (either sufficient or insufficient), the rest of the steps in the flowchart are skipped. In the example, WAGNERB does not own the CRLIMWRK file. The result of owner authority check is *No authority found* and checking continues.

6 Private authority is checked for the object. See Flowchart 4 on page 5-14.

7 If private authority (either sufficient or insufficient) is found for the profile, the rest of the steps in the flowchart are skipped. In the example, WAGNERB does not have any private authority to the CRLIMWRK file. The result is *No authority found* and checking continues.

8 The object is checked to see if it is secured by an authorization list. In the example, the CRLIMWRK file is secured by the CRLST1 authorization list.

9 If the object is secured by an authorization list, processing returns to step (4) with the authorization list as the object to check. In the example, the object to check is set to CRLST1.

10 If the object is not secured by an authorization list, the result is set to *No authority found.*

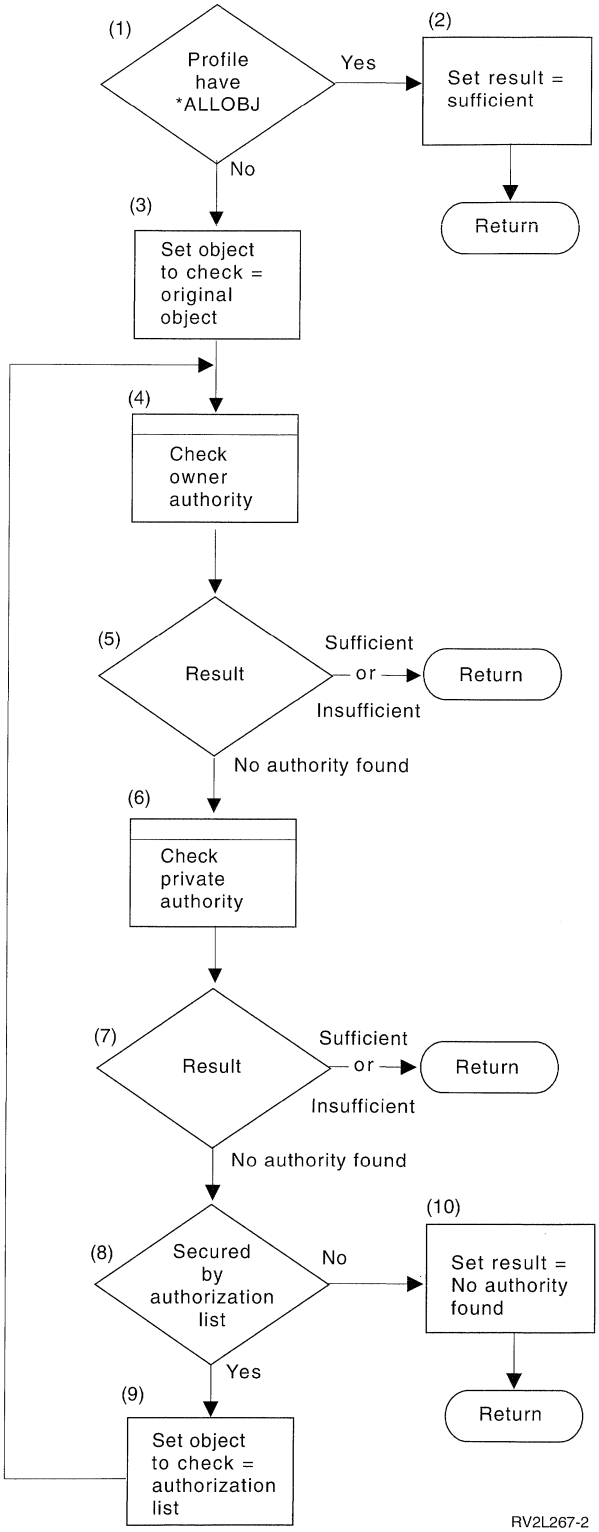


Figure 5-7. Flowchart 2: Check User Authority

Flowchart 3: How Owner Authority Is Checked:

Figure 5-8 shows the process for checking owner authority. The name of the owner profile and the owner's authority to an object are stored with the object.

Several possibilities exist for using owner authority to access an object:

- The user profile owns the object.
- The user profile owns the authorization list.
- The user's group profile owns the object.
- The user's group profile owns the authorization list.
- Adopted authority is used, and the program owner owns the object.
- Adopted authority is used, and the program owner owns the authorization list.

Following is the process used to check owner authority:

Step Description

1 The profile being tested is compared to the owner of the object (or authorization list). If they are the same, the owner's authority is checked. In the example, neither WAGNERB or DPTAR own the CRLIMWRK file.

2 When an object is created, the owner automatically has *ALL authority. However, the owner's authority can be changed or removed later.

3 If the profile being checked does not own the object or if the owner has no authority, the result is set to *No authority found* and the remaining steps are skipped.

4 The owner's authority is checked. The result is set to

5 *Sufficient*

or

6 *Not sufficient.*

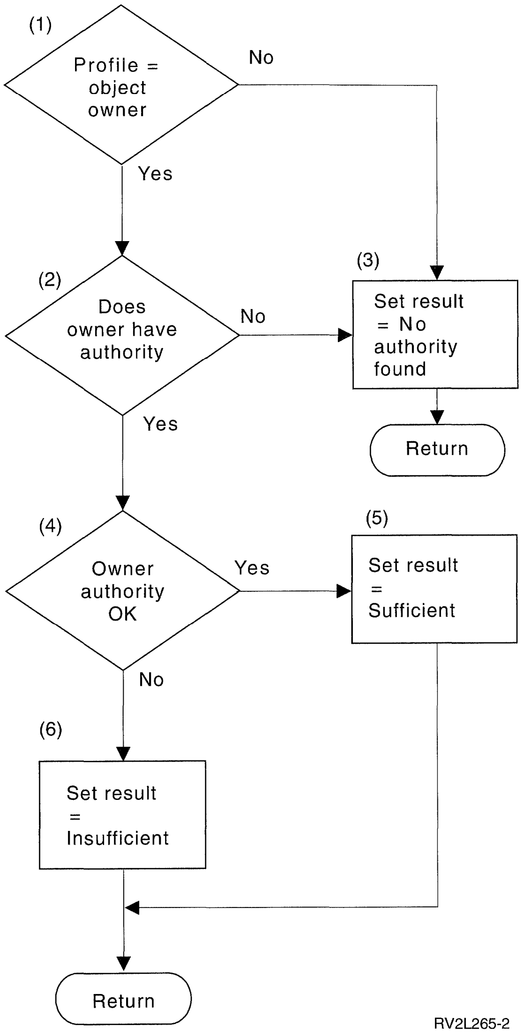


Figure 5-8. Flowchart 3: Owner Authority Checking

Flowchart 4: How Private Authority Is Checked:

Figure 5-9 on page 5-14 shows the process for checking private authorities. Private authority checking may be done using only information stored with the object, or it may require searching the private authorities stored with a user profile.

Step Description

1 The object is checked to see if any private authority exists. In the example, DPTMG and WILSONJ have private authority to the CRLIMWRK file.

2 If no private authority exists, the result is set to *No authority found* and the remaining steps in the flowchart are skipped.

3 The object is checked to see if any private authority exists that is less than public authority. The authority to the CRLIMWRK file in the example looks like this:

	---Object---			------Data------			
User	OPR	MGT	EXIST	READ	ADD	UPD	DLT
OWNAR		X					
DPTMG	X			X	X	X	X
WILSONJ							
*PUBLIC	X			X			

Authority is considered less than public if any authority that is present for *PUBLIC (in this case *OBJOPR and *READ) is not present for another user. In the example, the result of the test for the CRLIMWRK file is *Yes* because WILSONJ has *EXCLUDE authority. (OWNAR also has less authority than the public, but owner authority is not considered private authority.) The result of this test for the CRLST1 authorization list is *No*. (See Figure 5-5 on page 5-10.)

Steps (4) through (6) provide a method for using public authority, if possible, even though private authority exists for an object. The system tests to make sure that nothing later in the authority checking process might deny access to the object. If the result of these tests is *Sufficient*, searching private authorities can be avoided.

4 If no private authority is less than public authority, public authority is checked. In the example, public authority for the CRLIMWRK file is tested. It is not sufficient for the requested operation. (If public authority for an object is *AUTL, that is not sufficient for this test.)

5 If public authority is sufficient, owner authority is checked. If owner authority is not sufficient, additional authority checking must be done. The remaining steps cannot be bypassed because the user's group profile might own the object.

6 If the original object is secured by an authorization list, additional authority checking must be done. The system does not bypass the remaining steps because the user or group may have insufficient authority to the authorization list, or the group may have insufficient authority to the original object.

7 If all these tests are passed, the result is set to *Sufficient* and private authorities in the profile are not searched.

8 The private authorities of the profile are searched to determine if the user has authority to the object being checked. The results of the search can be *Sufficient*, *Insufficient*, or *No authority found*. **Searching private authorities is the most time-consuming part of the authority checking process.**

In the example, the steps in this chart are repeated for four different combinations, each requiring a search of private authorities in the profile:

Profile	Object	Type	Result
WAGNERB	CRLIMWRK	*FILE	No authority found
WAGNERB	CRLST1	*AUTL	No authority found
DPTAR	CRLIMWRK	*FILE	No authority found
DPTAR	CRLST1	*AUTL	Insufficient authority

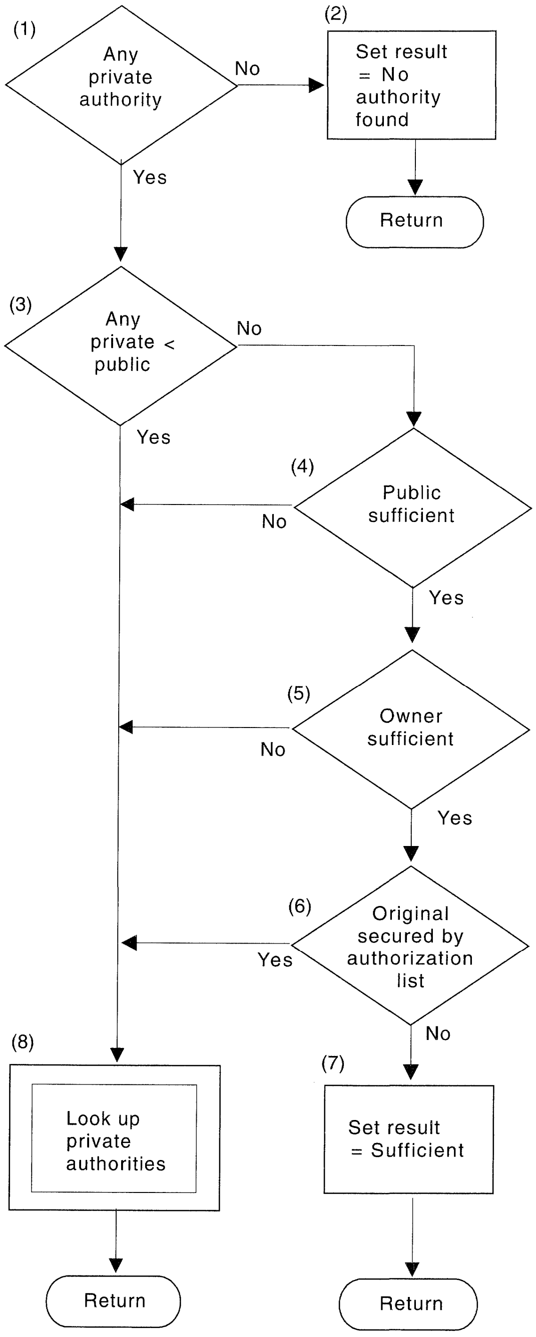


Figure 5-9. Flowchart 4: Private Authority Checking

Flowchart 5: How Public Authority Is Checked

Step Description

1 The public authority of the original object is checked to see if the object or its authorization list (public = *AUTL) should be used.

2 If the authorization list should be used (*AUTL), the object to be checked is set to the authorization list name.

3 If the authorization list is not used, the object to be checked is set to the object name.

4 Public authority is checked. The result is set to

5 *Sufficient*

or

6 *Insufficient.*

In the example, public authority is never checked because DPTAR has explicit, but insufficient, authority to the CRLST1 authorization list.

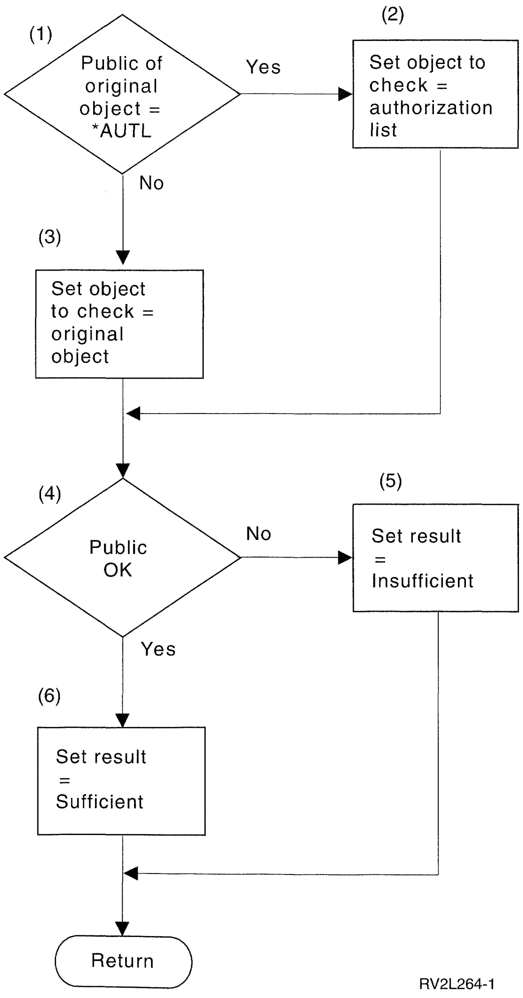

(1)
Public of original object = *AUTL
Yes
(2)
Set object to check = authorization list
No
(3)
Set object to check = original object
(4)
Public OK
No
(5)
Set result = Insufficient
Yes
(6)
Set result = Sufficient
Return
RV2L264-1

Figure 5-10. Flowchart 5: Check Public Authority

Flowchart 6: How Adopted Authority Is Checked:

If insufficient authority is found by checking user authority, the system checks adopted authority. The system may use adopted authority from the original program the user called or from earlier programs in the program stack. To provide the best performance and minimize the number of times private authorities are searched, the process for checking adopted authority looks first at *ALLOBJ special authority and owner authority for each program in the stack that uses adopted authority. If sufficient authority is not found, private authorities are searched for each program that uses adopted authority.

A new field, *Adopted authority check*, is used to keep track of the first and second times through the process. The *Result* field also has two additional values: *Continue* and *Pass 2*.

In the example to this point, WAGNERB has insufficient authority to perform the requested operation (CLRPFM) on the CRLIMWRK file. To continue the example, assume WAGNERB is using program ARPGM12. ARPGM12 is owned by DPTAR and uses adopted authority. The program ARPGM12 is called by program ARPGM01, which is owned by OWNAR and also adopts authority. The program stack for WAGNERB looks like this:

Program	Owner	Adopts
ARPGM12	DPTAR	Yes
ARPGM01	OWNAR	Yes

Step Description

1 These fields are set to begin the adopted authority checking process:

Field Name	Value
Object to check	Original object
Program to check	Original program
Adopted authority check	Pass 1

2 The first part of the adopted authority check is performed for the first program in the stack. (See Flowchart 7 on page 5-17.)

3 If the result is *Sufficient*, the remaining steps are skipped. In the example, when authority is checked for the owner of ARPGM01, the result is *Sufficient*, because OWNAR owns the CRLIMWRK file and has *OBJMGT authority to it.

4 If the result is *Insufficient* or *No authority found*, the program stack is checked to see whether previous programs use adopted authority. (See Flowchart 8 on page 5-18.)

In the example, when program ARPGM12 is checked, the result is *No authority found*, because DPTAR does not own the CRLIMWRK file.

5 If the result of the Use Adopted Authority Check is *Continue*, the first part of the adopted authority check is repeated for the next program in the stack, starting with step (2).

In the example, when authority is checked for ARPGM12, the result is *Continue*. The first part of the process is repeated using ARPGM01.

6 If the result is *Pass 2*, the second part of the adopted authority check is started. (See Flowchart 9 on page 5-19.)

7 If the result of the second adopted authority check is *Sufficient*, the remaining steps are skipped.

8 If the result is *Insufficient*, the program stack is checked to see whether previous programs use adopted authority. (See Flowchart 8 on page 5-18.)

9 If the result of the Use Adopted Authority check is *Sufficient* or *Insufficient*, the adopted authority checking process ends. If the result is *Continue*, the second part of the adopted authority checking process is repeated for the next program in the stack, starting with step (6).

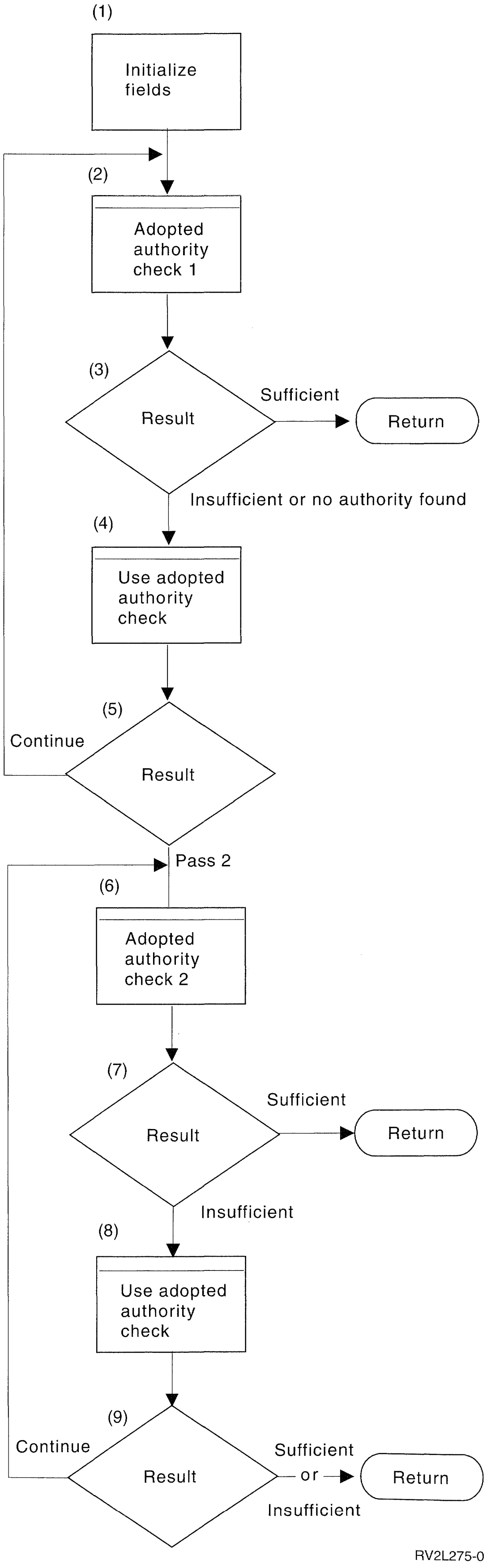


Figure 5-11. Flowchart 6: Check Adopted. Main Process.

Flowchart 7: Part 1 of Adopted Authority Checking: This flowchart describes the first part of the process for checking adopted authority. The system checks *ALLOBJ special authority and owner authority for each program that uses adopted authority.

Step Description

1 When checking authority, the system looks for the specific object and data authorities the user needs to perform the requested operation. When using adopted authority, authorities can be accumulated from more than one user profile. The system reduces the authority requested by the authority already found.

In the example, WAGNERB needs *OBJOPR, *OBJMGT, and *DLT authority to the CRLIMWRK file. Up to this point, WAGNERB has *OBJOPR and *DLT authority as a result of the authority DPTAR has to the CRLST1 authorization list. The adopted authority checking process needs to find only *OBJMGT authority.

2 Adopted authority is active for a program if the value of the USRPRF parameter for the program is *OWNER. In the example, the ARPGM12 program adopts the authority of its owner, DPTAR.

3 If the program does not adopt authority, the result is set to *Insufficient* and the remaining steps in this flowchart are skipped.

4 The *Profile to check* field is set to the name of the program owner. In the example, it is the DPTAR profile.

5 The profile is tested for *ALLOBJ special authority. In the example, neither DPTAR nor OWNAR has *ALLOBJ special authority.

6 If the profile has *ALLOBJ special authority, the result is *Sufficient*.

7 If the profile does not have *ALLOBJ special authority, the owner authority check is performed for the user profile. (See Flowchart 3 on page 5-13.)

In the example, this check is performed for both DPTAR (program ARPGM12) and OWNAR (ARPGM01). The result for DPTAR is *No authority found*. The result for OWNAR is *Sufficient*.

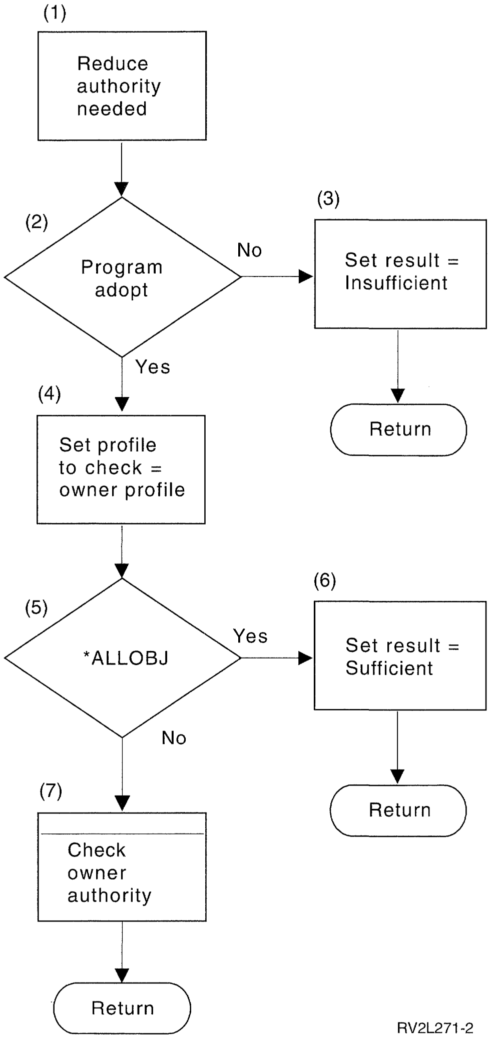


Figure 5-12. Flowchart 7: Adopted Authority Check Part 1

Flowchart 8: Using Adopted Authority from Previous Programs: This flowchart describes how the system determines whether to use adopted authority from previous programs in the program stack.

Step Description

1 The use adopted authority (USEADPAUT) parameter on a program determines whether adopted authority from any previous programs in the program stack can be used. In the example, the USEADPAUT parameter of ARPGM12 is *YES.

2 If USEADPAUT is *YES, the system checks for more programs in the program stack. In the example, another program, ARPGM01, is in the program stack.

3 If another program is found in the program stack, the *Program to check* field is set to that program name. In the example, the *Program to check* field is ARPGM01.

4 The *Object to check* is set to the original object. In the example, it is the CRLIMWRK file.

5 The result is set to *Continue* and adopted authority checking continues using the next program in the program stack.

6 If USEADPAUT is *NO or no more programs are in the program stack, the *Adopted authority check* field is tested.

7 If its value is *Pass 1*, fields are initialized to begin the second part of the adopted authority check:

Field Name	Value
Object to check	Original object
Program to check	Original program
Adopted authority check	Pass 2

8 The result is set to *Pass 2*, and the second part of the adopted authority checking process is started.

9 If no more programs are available to check in the second pass, the result is set to *Insufficient*.

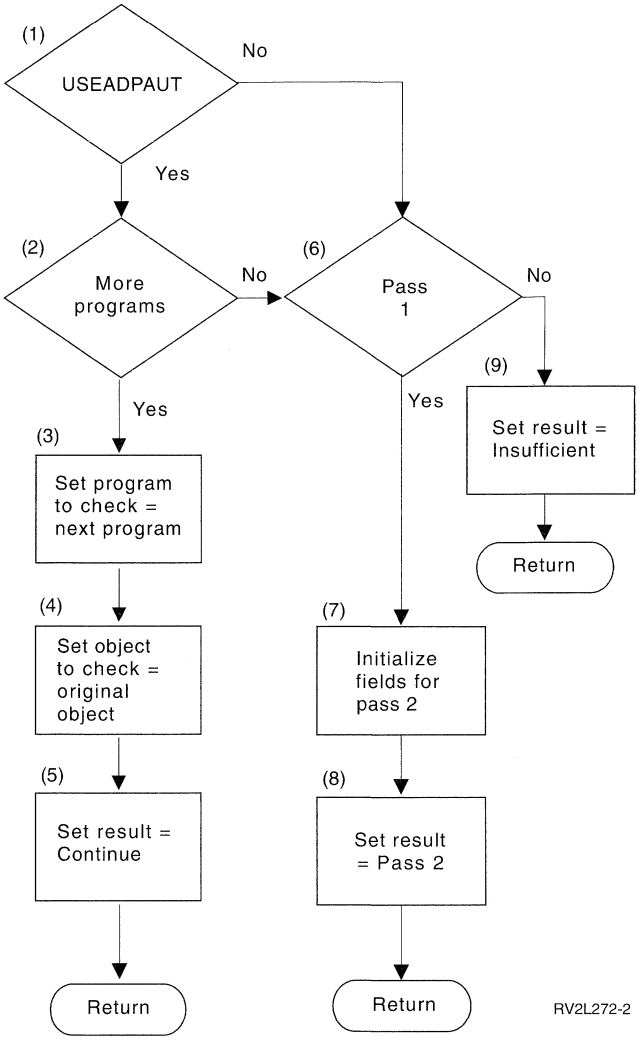


Figure 5-13. Flowchart 8: Use Adopted Authority Check

Flowchart 9: Part 2 of Adopted Authority Checking: The second part of the process for checking adopted authority looks up private authorities for each program that uses adopted authority.

Step Description

1 Adopted authority is active for a program if the value of the USRPRF parameter for the program is *OWNER.

2 If the program does not adopt authority, the result is set to *Insufficient* and the remaining steps in this flowchart are skipped.

3 The *Profile to check* field is set to the name of the program owner.

4 When checking authority, the system looks for the specific object and data authorities the user needs to perform the requested operation. When using adopted authority, authorities can be accumulated from more than one user profile. The system reduces the authority requested by the authority already found.

5 Private authorities are searched for the user profile. (See Flowchart 4 on page 5-14 for how this is done.)

6 If the result is *Sufficient*, the remaining steps are skipped.

7 If the result is *Insufficient* or *No authority found*, the object is checked to see whether it is secured by an authorization list.

8 If the object is not secured by an authorization list, the result is set to *Insufficient*, and processing continues.

9 If the object is secured by an authorization list, the *Object to check* field is set to the authorization list name.

10 The owner authority checking process is performed for the authorization list. (See Flowchart 3 on page 5-13.)

11 If the result is *Sufficient*, processing continues with the next step on Flowchart 6. If the result of the owner check is *Insufficient* or *No authority found*, private authority to the authorization list is checked, beginning with step (4).

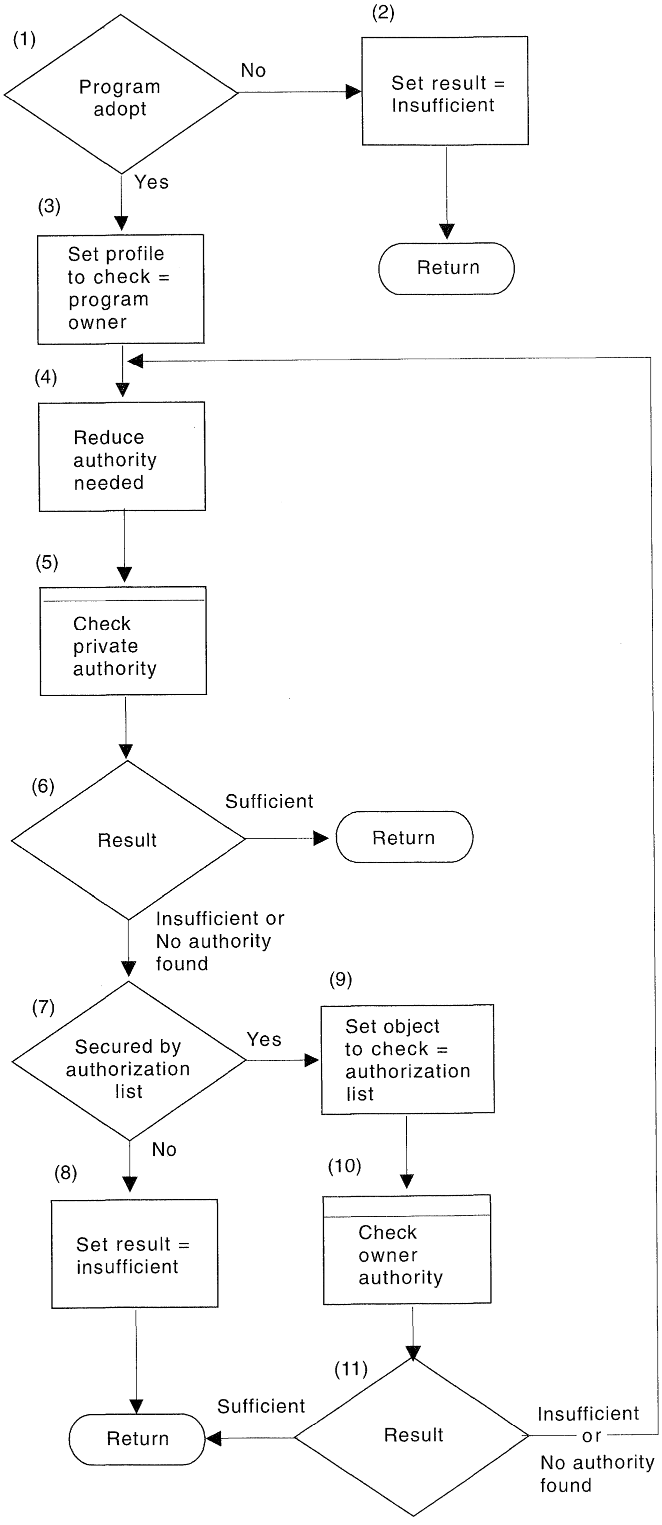

(1)
Program adopt
No
(2)
Set result = Insufficient
Return
Yes
(3)
Set profile to check = program owner
(4)
Reduce authority needed
(5)
Check private authority
(6)
Result
Sufficient
Return
Insufficient or No authority found
(7)
Secured by authorization list
Yes
(9)
Set object to check = authorization list
No
(8)
Set result = insufficient
(10)
Check owner authority
(11)
Result
Sufficient
Return
Insufficient or No authority found
RV2L273-1

Figure 5-14. Flowchart 9: Part 2 of Adopted Authority Checking

Summary of the Flowchart Example: Following is a summary of the steps required to complete the authority checking for the CRLIMWRK file used as the example in explaining the flowcharts. Authority for the file and the CRLST1 authorization list are shown in figures 5-4 and 5-5.

User WAGNERB is a member of group DPTAR. Neither has *ALLOBJ special authority. WAGNERB needs *OBJOPR, *OBJMGT, and *DLT authority to the CRLIMWRK file. WAGNERB is using program ARPGM12, which adopts the authority of its owner, DPTAR. Program ARPGM12 is called by program ARPGM01, which adopts the authority of its owner, OWNAR. The USEADPAUT parameter of ARPGM12 is *YES.

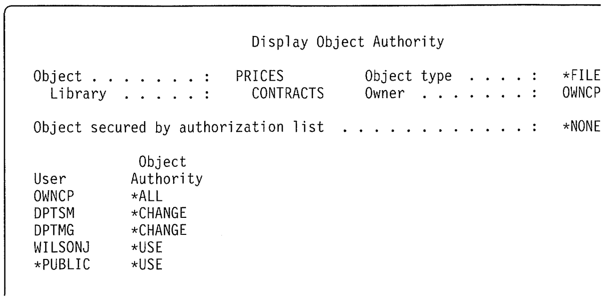
1. Flowchart 1, step 1. Profile to check = WAGNERB.
2. Flowchart 1, step 2.
   a. Flowchart 2, steps 1 and 3. Object to check = CUSTLIB/CRLIMWRK *FILE.
   b. Flowchart 2, step 4.
      1) Flowchart 3, steps 1 and 3. Result = No authority found.
   c. Flowchart 2, steps 5 and 6.
      1) Flowchart 4, steps 1, 3, and 8 (**first search of private authorities**). Result = No authority found.
   d. Flowchart 2, steps 7, 8 and 9. Object to check = CRLST1 *AUTL.
   e. Flowchart 2, step 4.
      1) Flowchart 3, steps 1 and 3. Result = No authority found.
   f. Flowchart 2, steps 5 and 6.
      1) Flowchart 4, steps 1, 3, and 4. Public is not sufficient.
      2) Flowchart 4, step 8 (**second search of private authorities**). Result = No authority found.
   g. Flowchart 2, steps 7, 8, and 10. Result = No authority found.
3. Flowchart 1, steps 3, 4, and 5. Profile to check = DPTAR.
4. Flowchart 1, step 2.
   a. Flowchart 2, steps 1 and 3. Object to check = CUSTLIB/CRLIMWRK *FILE.
   b. Flowchart 2, step 4.
      1) Flowchart 3, steps 1 and 3. Result = No authority found.
   c. Flowchart 2, steps 5 and 6.
      1) Flowchart 4, steps 1, 3, and 8 (**third search of private authorities**). Result = No authority found.
   d. Flowchart 2, steps 7, 8 and 9. Object to check = CRLST1 *AUTL.
   e. Flowchart 2, step 4.
      1) Flowchart 3, steps 1 and 3. Result = No authority found.
   f. Flowchart 2, steps 5 and 6.
      1) Flowchart 4, steps 1, 3 and 4. Public authority is not sufficient.
      2) Flowchart 4, step 8 (**fourth search of private authorities**). Result = Insufficient authority. (DPTAR has authority but does not have *OBJMGT.)
   g. Flowchart 2, step 7.
5. Flowchart 1, steps 3 and 8.
   a. Flowchart 6, step 1. Object to check = CUSTLIB/CRLIMWRK *FILE. Program to check = ARPGM12. Adopted authority check = Pass 1.
   b. Flowchart 6, step 2.
      1) Flowchart 7, step 1. Authority needed = *OBJMGT.
      2) Flowchart 7, steps 2 and 4. Profile to check = DPTAR.
      3) Flowchart 7, steps 5 and 7.
         a) Flowchart 3, steps 1 and 3. Result = No authority found.
   c. Flowchart 6, steps 3 and 4.
      1) Flowchart 8, steps 1, 2, and 3. Program to check = ARPGM01.
      2) Flowchart 8, step 4. Object to check = CUSTLIB/CRLIMWRK *FILE.
      3) Flowchart 8, step 5. Result = Continue.
   d. Flowchart 6, steps 5 and 2.
      1) Flowchart 7, steps 1, 2, and 4. Profile to chec = OWNAR.
      2) Flowchart 7, steps 5 and 7.
         a) Flowchart 3, Steps 1, 2, 4 and 5. Result = Sufficient. (OWNAR has *OBJMGT.)
   e. Flowchart 6, step 3.
6. Flowchart 1, step 9. Authorized.

Result: WAGNERB is authorized to perform the requested operation using a combination of DPTAR's authority to the CRLST1 authorization list and adopted authority.

Analysis: This example demonstrates most of the possibilities of authority checking. This example also demonstrates poor authority design, both from a management and performance standpoint. Too many options are used, making it di ficult to understand, change, and audit. Private authorities are searched four separate times, which may cause noticeable performance problems.

Examples of Authority Checking

Figure 5-15 on page 5-21 shows the authorities for the PRICES file. Following the figure are several examples of requested access to this file and the authority checking process. In the examples, searching private authorities (Flowchart 4, step 8) is highlighted, because this is the part of the authority checking process that can cause performance problems if it is repeated several times.



```
                          Display Object Authority

Object . . . . . . . :   PRICES       Object type  . . . . :   *FILE
  Library  . . . . . :    CONTRACTS   Owner  . . . . . . . :   OWNCP

Object secured by authorization list  . . . . . . . . . . . . :   *NONE

            Object
User        Authority
OWNCP       *ALL
DPTSM       *CHANGE
DPTMG       *CHANGE
WILSONJ     *USE
*PUBLIC     *USE
```

Figure 5-15. Authority for Prices File

Case 1: Using Group Authority: User ROSSM wants to access the PRICES file using the program CPPGM01. CPPGM01 requires *CHANGE authority to the file. ROSSM is a member of group profile DPTSM. Neither ROSSM nor DPTSM has *ALLOBJ special authority. The system performs these steps in determining whether to allow ROSSM access to the PRICES file:

1. Flowchart 1, step 1. Profile to check = ROSSM.
2. Flowchart 1, step 2.
   a. Flowchart 2, steps 1 and 3. Object to check = CONTRACTS/PRICES *FILE.
   b. Flowchart 2, step 4.
      1) Flowchart 3, steps 1 and 3. Result = No authority found.
   c. Flowchart 2, steps 5 and 6.
      1) Flowchart 4, steps 1, 3, and 4. Public is not sufficient.
      2) **Flowchart 4, step 8**. Result = No authority found.
   d. Flowchart 2, steps 7, 8, and 10. Result = No authority found. (The prices file is not secured by an authorization list.)
3. Flowchart 1, steps 3, 4, and 5. Profile to check = DPTSM.
4. Flowchart 1, step 2.
   a. Flowchart 2, steps 1 and 3. Object to check = CONTRACTS/PRICES *FILE.
   b. Flowchart 2, step 4.
      1) Flowchart 3, steps 1 and 3. Result = No authority found.
   c. Flowchart 2, steps 5 and 6.
      1) Flowchart 4, steps 1, 3, and 4. Public is not sufficient.
      2) **Flowchart 4, step 8**. Result = Sufficient. (DPTSM has *CHANGE authority.)
   d. Flowchart 2, step 7.
5. Flowchart 1, step 3. Authorized.

Result: ROSSM is authorized because the group profile DPTSM has *CHANGE authority.

Analysis: Using group authority in this example is a good method for managing authorities. It reduces the number of private authorities on the system and is easy to understand and audit. However, using group authority usually causes two searches of private authorities (for the user and the group), when public authority is not adequate. Avoid using group authority when it does not provide significant benefits in managing authority.

Case 2: Using Public Authority: User JONESP wants to access the PRICES file using the program CPPGM06. CPPGM06 requires *USE authority to the file. JONESP is a member of group profile DPTSM and does not have *ALLOBJ special authority. The system performs these steps in determining whether to allow JONESP access to the PRICES file:

1. Flowchart 1, step 1. Profile to check = JONESP.
2. Flowchart 1, step 2.
   a. Flowchart 2, steps 1 and 3. Object to check = CONTRACTS/PRICES *FILE.
   b. Flowchart 2, step 4.
      1) Flowchart 3, steps 1 and 3. Result = No authority found.
   c. Flowchart 2, steps 5 and 6.
      1) Flowchart 4, steps 1, 3, and 4. Public is sufficient.
      2) Flowchart 4, step 5. Owner authority is sufficient. (OWNCP has *ALL.)
      3) Flowchart 4, steps 6 and 7. Result = Sufficient.
   d. Flowchart 2, step 7.
3. Flowchart 1, step 3. Authorized.

Analysis: This example shows the performance benefit gained when you avoid defining any private authority less than public authority and make public authority sufficient for some application functions. Although private authority exists for the PRICES file, the public authority is sufficient for this request and can be used without searching private authorities.

Case 3: Using Adopted Authority: User SMITHG wants to access the PRICES file using program CPPGM08. SMITHG is not a member of a group and does not have *ALLOBJ special authority. Program CPPGM08 requires *CHANGE authority to the file. CPPGM08 is owned by the profile OWNCP and adopts owner authority (USRPRF is *OWNER).

1. Flowchart 1, step 1. Profile to check = SMITHG.
2. Flowchart 1, step 2.
   a. Flowchart 2, steps 1 and 3. Object to check = CONTRACTS/PRICES *FILE.
   b. Flowchart 2, step 4.
      1) Flowchart 3, steps 1 and 3. Result = No authority found.
   c. Flowchart 2, steps 5 and 6.
      1) Flowchart 4, steps 1, 3, and 4. Public is not sufficient.
      2) **Flowchart 4, step 8**. Result = No authority found.
   d. Flowchart 2, steps 7, 8, and 10. Result = No authority found.
3. Flowchart 1, steps 3, 4, and 6.
   a. Flowchart 5, steps 1 and 3. Object to check = CONTRACTS/PRICES *FILE.

   b. Flowchart 5, steps 4 and 5. Result = Insufficient authority.
4. Flowchart 1, steps 7 and 8.
   a. Flowchart 6, step 1. Object to check = CONTRACTS/PRICES *FILE. Program to check = CPPGM08. Adopted authority check = Pass 1.
   b. Flowchart 6, step 2.
      1) Flowchart 7, step 1. Because the public has *USE authority (*OBJOPR and *READ), the authority still needed is *ADD, *UPD, and *DLT. (Original authority needed was *CHANGE.)
      2) Flowchart 7, steps 2 and 4. Profile to check = OWNCP.
      3) Flowchart 7, steps 5 and 7.
         a) Flowchart 3, steps 1, 2, 4, and 5. Result = Sufficient.
   c. Flowchart 6, step 3.
5. Flowchart 1, step 9. Authorized.

Analysis: This example demonstrates the performance advantage in using adopted authority when the program owner also owns the application objects.

The number of steps required to perform authority checking has almost no impact on performance, because most of the steps do not require retrieving new information. In this example, although many steps are performed, private authorities are searched only once (for user SMITHG).

Compare this with Case 1 on page 5-21. If you were to change Case 1 so that the group profile DPTSM owns the PRICES file and has *ALL authority to it, the performance characteristics of the two examples would be the same. However, having a group profile own application objects may represent a security exposure. The members of the group always have the group's (owner) authority, unless you specifically give group members less authority. When you use adopted authority, you can control the situations in which owner authority is used.

Case 4: User and Group Authority: User WILSONJ wants to access file PRICES using program CPPGM01, which requires *CHANGE authority. WILSONJ is a member of group profile DPTSM and does not have *ALLOBJ special authority. Program CPPGM01 does not use adopted authority, and it ignores any previous adopted authority (USEADPAUT is *NO).

1. Flowchart 1, step 1. Profile to check = WILSONJ.
2. Flowchart 1, step 2.
   a. Flowchart 2, steps 1 and 3. Object to check = CONTRACTS/PRICES *FILE.
   b. Flowchart 2, step 4.
      1) Flowchart 3, steps 1 and 3. Result = No authority found.
   c. Flowchart 2, steps 5 and 6.
      1) Flowchart 4, steps 1, 3, and 4. Public is not sufficient.
      2) **Flowchart 4, step 8**. Result = Insufficient authority.
   d. Flowchart 2, step 5.
3. Flowchart 1, steps 3 and 8.
   a. Flowchart 6, step 1. Object to check = CONTRACTS/PRICES *FILE. Program to check = CPPGM01. Adopted authority check = Pass 1.
   b. Flowchart 6, step 2.
      1) Flowchart 7, step 1. Authority needed is *ADD *UPD, and *DLT. (WILSONJ has *OBJOPR and *READ.)
      2) Flowchart 7, steps 2 and 3. Result = Insufficient. (Program CPPGM01 does not adopt authority.)
   c. Flowchart 6, steps 3 and 4.
      1) Flowchart 8, step 1. CPPGM01 ignores the adopted authority of previous programs (USEADPAUT = *NO).
      2) Flowchart 8, steps 6 and 7. Object to check = CONTRACTS/PRICES *FILE. Program to check = CPPGM01. Adopted authority check = Pass 2.
      3) Flowchart 8, step 8. Result = Pass 2.
   d. Flowchart 6, steps 5 and 6.
      1) Flowchart 9, steps 1 and 2. Result = Insufficient.
   e. Flowchart 6, steps 7 and 8.
      1) Flowchart 8, steps 1, 6, and 9. Result = Insufficient.
   f. Flowchart 6, step 9. Result = Insufficient.
4. Flowchart 1, step 9. Not authorized.

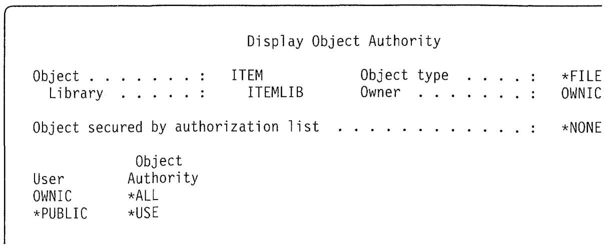
Analysis: This example demonstrates that a user can be denied access to an object even though the user's group ha sufficient authority.

Giving a user the same authority as the public but less than the user's group does not affect the performance of authorit checking for other users. However, if WILSONJ had *EXCLUDE authority (less than public), you would lose the performance benefits shown in Case 2.

Although this example has many steps, private authorities are searched only once. This should provide acceptable performance.

Case 5: Public Authority without Private

Authority: The authority information for the ITEM file look like this:



```
                         Display Object Authority

Object . . . . . . . :   ITEM         Object type  . . . . :   *FILE
  Library  . . . . . :     ITEMLIB    Owner  . . . . . . . :   OWNIC

Object secured by authorization list  . . . . . . . . . . . :   *NONE

              Object
User          Authority
OWNIC         *ALL
*PUBLIC       *USE
```

If WILSONJ (or any other user) needs *USE authority to the ITEM file, these are the authority-checking steps:

1. Flowchart 1, step 1. Profile to check = WILSONJ.
2. Flowchart 1, step 2.
    a. Flowchart 2, step 1 and 3. Object to check = ITEMLIB/ITEM *FILE.
    b. Flowchart 2, step 4.
        1) Flowchart 3, steps 1 and 3. Result = No authority found.
    c. Flowchart 2, steps 5 and 6.
        1) Flowchart 4, steps 1 and 2. Result = No authority found.
    d. Flowchart 2, steps 7, 8, and 10. Result = No authority found.
3. Flowchart 1, steps 4 and 6. (If the user is a group, steps 2 through 2d would be repeated for the group.)
    a. Flowchart 5, steps 1 and 3. Object to check = ITEMLIB/ITEM *FILE
    b. Flowchart 5, steps 4 and 6. Result = Sufficient authority.
4. Flowchart 1, step 7. Authorized.

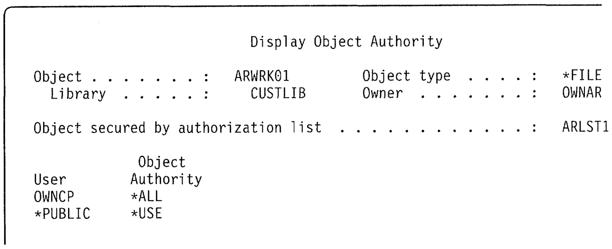
Analysis: Public authority provides the best performance when it is used without any private authorities. In this example, private authorities are never searched.

Case 6: Adopted Authority without Private Authority: For this example, all programs in the application are owned by the OWNIC profile. Any program in the application requiring more than *USE authority adopts owner authority. These are the steps for user WILSONJ to obtain *CHANGE authority to the ITEM file using program ICPGM10, which adopts authority:

1. Flowchart 1, step 1. Profile to check = WILSONJ.
2. Flowchart 1, step 2.
    a. Flowchart 2, steps 1 and 3. Object to check = ITEMLIB/ITEM *FILE.
    b. Flowchart 2, step 4.
        1) Flowchart 3, steps 1 and 3. Result = No authority found.
    c. Flowchart 2, steps 5 and 6.
        1) Flowchart 4, steps 1 and 2. Result = No authority found.
    d. Flowchart 2, steps 7, 8, and 10. Result = No authority found.
3. Flowchart 1, steps 4 and 6.
    a. Flowchart 5, steps 1 and 3. Object to check = ITEMLIB/ITEM *FILE
    b. Flowchart 5, steps 4 and 5. Result = Insufficient authority.
4. Flowchart 1, steps 7 and 8.
    a. Flowchart 6, step 1. Object to check = ITEMLIB/ITEM *FILE. Program to check = ICPGM10. Adopted authority check = Pass 1.
    b. Flowchart 6, step 2.
        1) Flowchart 7, step 1. Authority needed is reduced to *ADD, *UPD, and *DLT, because public authority is *USE (*OBJOPR and *READ).
        2) Flowchart 7, steps 2 and 4. Profile to check = OWNIC.
        3) Flowchart 7, steps 5 and 7.
            a) Flowchart 3, steps 1, 2, 4 and 5. Result = Sufficient.
    c. Flowchart 6, step 3.
5. Flowchart 1, step 9. Authorized.

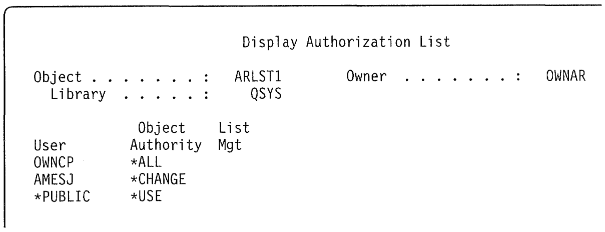
Analysis: This example shows the benefits of using adopted authority without private authority, particularly if the owner of the programs also owns application objects. This example did not require searching private authorities.

Case 7: Using an Authorization List: The ARWKR01 file in library CUSTLIB is secured by the ARLST1 authorization list. Figure 5-16 and Figure 5-17 show the authorities:



```
                        Display Object Authority

Object . . . . . . . :   ARWRK01        Object type . . . . :   *FILE
  Library . . . . . :     CUSTLIB       Owner . . . . . . . :   OWNAR

Object secured by authorization list . . . . . . . . . . . . :   ARLST1

            Object
User        Authority
OWNCP       *ALL
*PUBLIC     *USE
```

Figure 5-16. Authority for the ARWRK01 File



```
                        Display Authorization List

Object . . . . . . . :   ARLST1        Owner . . . . . . . :   OWNAR
  Library . . . . . :      QSYS

            Object    List
User        Authority Mgt
OWNCP       *ALL
AMESJ       *CHANGE
*PUBLIC     *USE
```

Figure 5-17. Authority for the ARLST1 Authorization List

User AMESJ, who is not a member of a group profile, needs *CHANGE authority to the ARWRK01 file. These are the authority-checking steps:

1. Flowchart 1, step 1. Profile to check = AMESJ.
2. Flowchart 1, step 2.
    a. Flowchart 2, steps 1 and 3. Object to check = CUSTLIB/ARWRK01 *FILE.
    b. Flowchart 2, step 4.
        1) Flowchart 3, steps 1 and 3. Result = No authority found.
    c. Flowchart 2, steps 5 and 6.
        1) Flowchart 4, steps 1 and 2. Result = No authority found.
    d. Flowchart 2, steps 7, 8 and 9. Object to check = ARLST1 *AUTL.
    e. Flowchart 2, step 4.
        1) Flowchart 3, steps 1 and 3. Result = No authority found.
    f. Flowchart 2, steps 5 and 6.
        1) Flowchart 4, steps 1, 3 and 4. Public authority is not sufficient.

2) **Flowchart 4, step 8**. Result = Sufficient.

g. Flowchart 2, step 7.

3. Flowchart 1, step 3. Authorized.

Analysis: This example demonstrates that authorization lists can make authorities easy to manage and provide good performance. This is particularly true if objects secured by the authorization list do not have any private authorities.

If AMESJ were a member of a group profile, it would add additional steps to this example, but it would not add an additional search of private authorities, as long as no private authorities are defined for the ARWRK01 file. Performance problems are most likely to occur when an object is secured by an authorization list and has private authorities, and when a user seeking access is a member of group profile, as shown in the CRLIMWRK file (see Figure 5-4 on page 5-10).

Working with Authority

This part of the chapter describes commonly-used methods for setting up, maintaining, and displaying authority information on your system. Appendix A provides a complete list of the commands available for working with authority. The descriptions that follow do not discuss all the parameters for commands or all the fields on the displays. Consult online information for complete details.

Working with Libraries

Two parameters on the Create Library (CRTLIB) command affect authority:

Authority (AUT): The AUT parameter can be used to specify the public authority for the library or the authorization list that secures the library. The AUT parameter applies to the library itself, not to the objects in the library. If you specify an authorization list name, the public authority for the library is set to *AUTL.

If you do not specify AUT when you create a library, *LIBCRTAUT is the default. The system uses the CRTAUT value from the QSYS library, which is shipped as *SYSVAL.

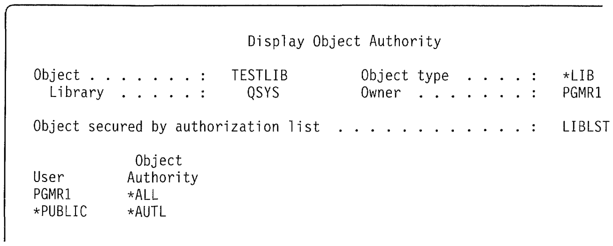
Create Authority (CRTAUT): The CRTAUT parameter determines the default authority for any new objects that are created in the library. CRTAUT can be set to one of the system-defined authorities (*ALL, *CHANGE, *USE, or *EXCLUDE), to *SYSVAL (the QCRTAUT system value), or to the name of an authorization list.

Note: You can change the CRTAUT value for a library using the Change Library (CHGLIB) command.

If user PGMR1 enters this command:

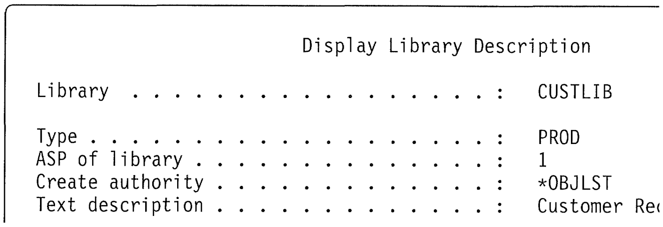
```
CRTLIB TESTLIB AUT(LIBLST) CRTAUT(OBJLST)
```

the authority for the library looks like this:



```
                         Display Object Authority

Object . . . . . . . :   TESTLIB        Object type  . . . . :   *LIB
  Library  . . . . . :     QSYS         Owner  . . . . . . . :   PGMR1

Object secured by authorization list  . . . . . . . . . . . . :   LIBLST

           Object
User       Authority
PGMR1      *ALL
*PUBLIC    *AUTL
```

- Because an authorization list was specified for the AUT parameter, public authority is set to *AUTL.
- The user entering the CRTLIB command owns the library, unless the user's profile specifies OWNER(GRPPRF). The owner is automatically given *ALL authority.
- The CRTAUT value is not shown on the object authority displays. Use the Display Library Description (DSPLIBD) command to see the CRTAUT value for a library.



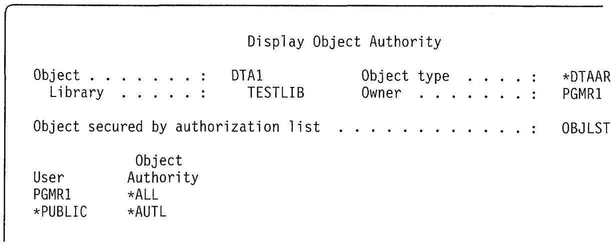
```
                         Display Library Description

Library  . . . . . . . . . . . . . . . . . :   CUSTLIB

Type . . . . . . . . . . . . . . . . . . . :   PROD
ASP of library . . . . . . . . . . . . . . :   1
Create authority . . . . . . . . . . . . . :   *OBJLST
Text description . . . . . . . . . . . . . :   Customer Re
```

Creating Objects

When you create a new object, you can either specify the authority (AUT) or use the default, *LIBCRTAUT. If PGMR1 enters this command:

```
CRTDTAARA (TESTLIB/DTA1) +
   TYPE(*CHAR)
```

the authority for the data area looks like this:



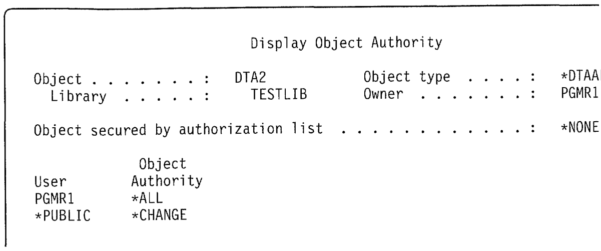
```
                         Display Object Authority

Object . . . . . . . :   DTA1           Object type  . . . . :   *DTAARA
  Library  . . . . . :     TESTLIB      Owner  . . . . . . . :   PGMR1

Object secured by authorization list  . . . . . . . . . . . . :   OBJLST

           Object
User       Authority
PGMR1      *ALL
*PUBLIC    *AUTL
```

The authorization list (OBJLST) comes from the CRTAUT parameter that was specified when TESTLIB was created.

If PGMR1 enters this command:

```
CRTDTAARA (TESTLIB/DTA2) AUT(*CHANGE) +
   TYPE(*CHAR)
```

the authority for the data area looks like this:



```
                     Display Object Authority

Object . . . . . . . :   DTA2          Object type  . . . . :   *DTAARA
  Library  . . . . . :     TESTLIB     Owner  . . . . . . . :   PGMR1

Object secured by authorization list  . . . . . . . . . . . . :   *NONE

              Object
User          Authority
PGMR1         *ALL
*PUBLIC       *CHANGE
```

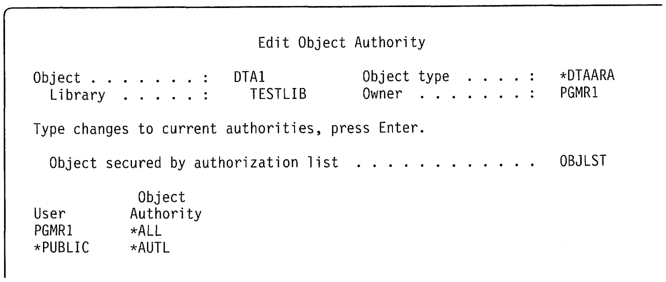
Working with Individual Object Authority

To change the authority for an object you must have one of the following:

- *ALLOBJ authority or membership in a group profile that has *ALLOBJ special authority.

  Note: The group's authority is not used if you have private authority to the object.
- Ownership of the object. If a group profile owns the object, any member of the group can act as the object owner, unless the member has been given specific authority that does not meet the requirements for changing the object's authority.
- *OBJMGT authority to the object and any authorities being granted or revoked (except *EXCLUDE). Any user who is allowed to work with the object's authority can grant or revoke *EXCLUDE authority.

The easiest way to change authority for an individual object is with the Edit Object Authority display. This display can be called directly by using the Edit Object Authority (EDTOBJAUT) command or selected as an option from the Work with Objects by Owner (WRKOBJOWN) or WRKOBJ (Work with Objects) display.

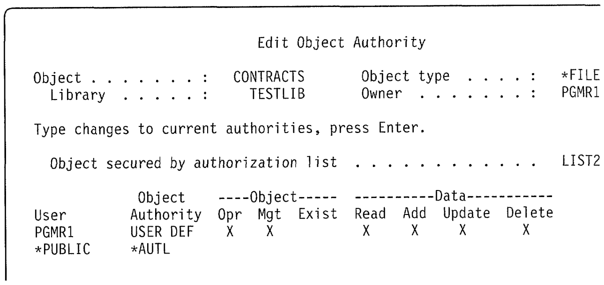


```
                        Edit Object Authority

Object . . . . . . . :   DTA1          Object type  . . . . :   *DTAARA
  Library  . . . . . :     TESTLIB     Owner  . . . . . . . :   PGMR1

Type changes to current authorities, press Enter.

  Object secured by authorization list  . . . . . . . . . . . .   OBJLST

              Object
User          Authority
PGMR1         *ALL
*PUBLIC       *AUTL
```

Specifying User-Defined Authority: The Object Authority column on the Edit Object Authority display allows you to specify any of the system-defined sets of authorities (*ALL, *CHANGE, *USE, *EXCLUDE). If you want to specify authority that is not a system-defined set, use F11 (Display detail).

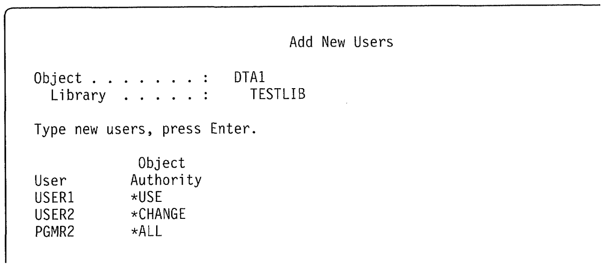
Note: If the *User options* (USROPT) field in your user profile is set to *EXPERT, you always see this detailed version of the display without having to press F11.

For example, PGMR1 removes *OBJEXIST authority to the CONTRACTS file, to prevent accidentally deleting the file. Because PGMR1 has a combination of authorities that is not one of the system-defined sets, the system puts *USER DEF* (user-defined) in the Object Authority column:



```
                        Edit Object Authority

Object . . . . . . . :   CONTRACTS     Object type  . . . . :   *FILE
  Library  . . . . . :     TESTLIB     Owner  . . . . . . . :   PGMR1

Type changes to current authorities, press Enter.

  Object secured by authorization list  . . . . . . . . . . . .   LIST2

             Object    ----Object-----  ----------Data-----------
User         Authority  Opr  Mgt  Exist  Read  Add  Update  Delete
PGMR1        USER DEF    X    X           X     X     X       X
*PUBLIC      *AUTL
```

Giving Authority to New Users: To give authority to additional users, press F6 (Add new users) from the Edit Object Authority display. You see the Add New Users display, which allows you to define authority for multiple users:



```
                            Add New Users

Object . . . . . . . :   DTA1
  Library  . . . . . :     TESTLIB

Type new users, press Enter.

              Object
User          Authority
USER1         *USE
USER2         *CHANGE
PGMR2         *ALL
```

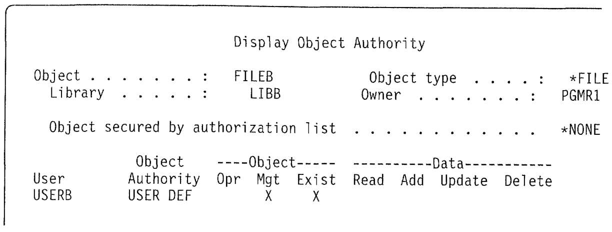
Removing a User's Authority: Removing a user's authority for an object is different from giving the user *EXCLUDE authority. *EXCLUDE authority means the user is specifically not allowed to use the object. Only *ALLOBJ special authority and adopted authority override *EXCLUDE authority. Removing a user's authority means the user has no specific authority to the object. The user can gain access through a group profile, an authorization list, public authority, *ALLOBJ special authority, or adopted authority.

You can remove a user's authority using the Edit Object Authority display. Type blanks in the Object Authority field for the user and press the Enter key. The user is removed from the display. You can also use the Revoke Object Authority (RVKOBJAUT) command. Either revoke the specific authority the user has or revoke *ALL authority for the user.

Note: The RVKOBJAUT command revokes only the authority you specify. For example, USERB has *ALL authority to FILEB in library LIBB. You revoke *CHANGE authority:

```
| RVKOBJAUT OBJ(LIBB/FILEB) OBJTYPE(*FILE) +
| USER(*USERB) AUT(*CHANGE)
```

| After the command, USERB's authority to FILEB looks like
| this:



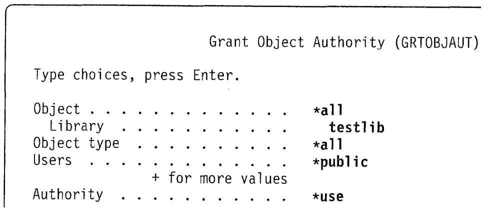
```
                      Display Object Authority

Object . . . . . . . :   FILEB         Object type  . . . . :    *FILE
  Library  . . . . . :     LIBB        Owner  . . . . . . . :    PGMR1

  Object secured by authorization list  . . . . . . . . . . . .  *NONE

              Object    ----Object-----  ----------Data-----------
User          Authority  Opr  Mgt  Exist  Read  Add  Update  Delete
USERB         USER DEF        X     X
```

Working with Authority for Multiple Objects

The Edit Object Authority display allows you to interactively work with the authority for one object at a time. The Grant Object Authority (GRTOBJAUT) command allows you to make authority changes to more than one object at a time. You can use the GRTOBJAUT authority command interactively or in batch. You can also call it from a program.

Following are examples of using the GRTOBJAUT command, showing the prompt display. When the command runs, you receive a message for each object indicating whether the change was made. Authority changes require an exclusive lock on the object and cannot be made when an object is in use. Print your job log for a record of changes attempted and made.

- To give all the objects in the TESTLIB library a public authority of *USE:



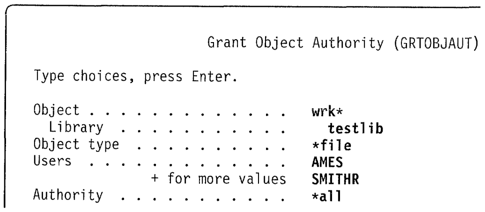
```
                Grant Object Authority (GRTOBJAUT)

Type choices, press Enter.

Object . . . . . . . . . . . . .   *all
  Library  . . . . . . . . . . .     testlib
Object type  . . . . . . . . . .   *all
Users  . . . . . . . . . . . . .   *public
               + for more values
Authority  . . . . . . . . . . .   *use
```

The GRTOBJAUT command gives the authority you specify, but it does not remove any authority that is greater than you specified. If some objects in the TESTLIB library have public authority *CHANGE, the command just shown would not reduce their public authority to *USE. To make sure that all objects in TESTLIB have a public authority of *USE, use the following sequence of commands:

```
RVKOBJAUT OBJ(TESTLIB/*ALL) OBJTYPE(*ALL) +
          USER(*PUBLIC) AUT(*ALL)
GRTOBJAUT OBJ(TESTLIB/*ALL) OBJTYPE(*ALL) +
          USER(*PUBLIC) AUT(*USE)
```

These commands set public authority only for objects that currently exist in the library. To set the public authority for any new objects that are created later, use the CRTAUT parameter on the library description.

- To give *ALL authority to the work files in the TESTLIB library to users AMES and SMITHR. In this example, work files all start with the characters WRK:

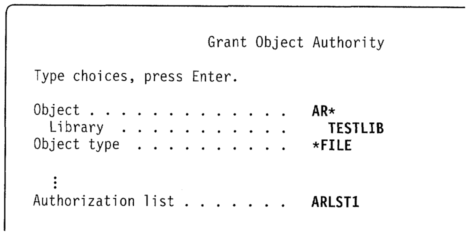


```
                Grant Object Authority (GRTOBJAUT)

Type choices, press Enter.

Object . . . . . . . . . . . . .   wrk*
  Library  . . . . . . . . . . .     testlib
Object type  . . . . . . . . . .   *file
Users  . . . . . . . . . . . . .   AMES
               + for more values   SMITHR
Authority  . . . . . . . . . . .   *all
```

This command uses a generic name to specify the files. You specify a generic name by typing a character string followed by an asterisk (*). Online information tells which parameters of a command allow a generic name.

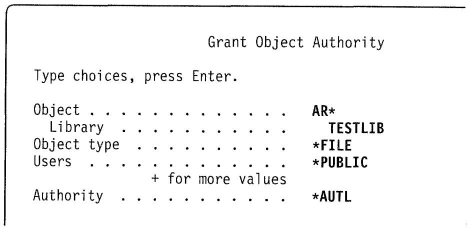
- To secure all the files starting with the characters AR* using an authorization list called ARLST1 and have the files get their public authority from the list, use the following two commands:

1. Secure the files with the authorization list using the GRTOBJAUT command:



```
                Grant Object Authority

Type choices, press Enter.

Object . . . . . . . . . . . . .   AR*
  Library  . . . . . . . . . . .     TESTLIB
Object type  . . . . . . . . . .   *FILE

  :
Authorization list . . . . . . .   ARLST1
```

2. Set public authority for the files to *AUTL, using the GRTOBJAUT command:

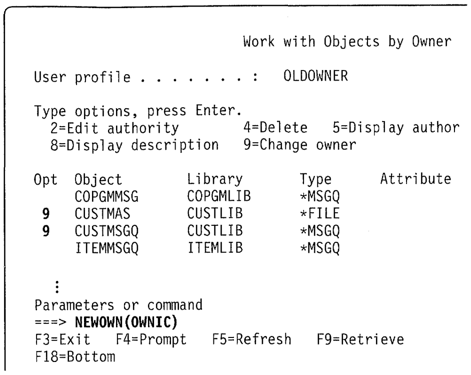


```
                Grant Object Authority

Type choices, press Enter.

Object . . . . . . . . . . . . .   AR*
  Library  . . . . . . . . . . .     TESTLIB
Object type  . . . . . . . . . .   *FILE
Users  . . . . . . . . . . . . .   *PUBLIC
               + for more values
Authority  . . . . . . . . . . .   *AUTL
```

Working with Object Ownership

To change ownership of an object, use the Change Object Owner (CHGOBJOWN) command or the Work with Objects by Owner (WRKOBJOWN) command.

The Work with Objects by Owner display shows all the objects owned by a profile. You can assign individual objects to a new owner. You can also change ownership for more than one object at a time by using the NEWOWN (new owner) parameter at the bottom of the display:



```
                          Work with Objects by Owner

User profile . . . . . . . :   OLDOWNER

Type options, press Enter.
  2=Edit authority         4=Delete   5=Display author
  8=Display description    9=Change owner

Opt  Object        Library       Type       Attribute
     COPGMMSG      COPGMLIB      *MSGQ
 9   CUSTMAS       CUSTLIB       *FILE
 9   CUSTMSGQ      CUSTLIB       *MSGQ
     ITEMMSGQ      ITEMLIB       *MSGQ

  :
Parameters or command
===> NEWOWN(OWNIC)
F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve
F18=Bottom
```

When you change ownership using either method, you can choose to remove the previous owner's authority to the object. The default for the CUROWNAUT (current owner authority) parameter is *REVOKE.

To transfer ownership of an object, you must have:

- Object existence authority for the object
- *ALL authority or ownership, if the object is an authorization list
- Add authority for the new owner's user profile
- Delete authority for the present owner's user profile

The Change Library Owner (CHGLIBOWN) and Check Library Owner (CHKLIBOWN) commands in the QUSRTOOL library can help you analyze and manage the ownership of objects in your libraries.

You cannot delete a user profile that owns objects. The topic "Deleting User Profiles" on page 4-20 shows methods for handling owned objects when deleting a profile.

Using a Referenced Object

Both the Edit Object Authority display and the GRTOBJAUT command allow you to give authority to an object (or group of objects) based on the authority of a referenced object. This is a useful tool in some situations, but you should also evaluate the use of an authorization list to meet your requirements. See the "Authorization Lists and Referenced Objects" on page 7-10 for a comparison of the two.

Copying Authority from a User

You can copy all the private authorities from one user profile to another using the Grant User Authority (GRTUSRAUT) command. This method can be useful in certain situations. For example, the system does not allow you to rename a user profile. To create an identical profile with a different name involves several steps, including copying the original profile's authorities. "Renaming a User Profile" on page 4-22 shows an example of how to do this.

The GRTUSRAUT command copies private authorities only. It does not copy special authorities, nor does it transfer object ownership.

The GRTUSRAUT command should not be used in place of creating group profiles. GRTUSRAUT creates a duplicate set of private authorities, which increases the time it takes to save the system and makes authority management more difficult. GRTUSRAUT copies authorities as they exist at a particular moment. If authority is required to new objects in the future, each profile must be granted authority individually. The group profile provides this function automatically.

To use the GRTUSRAUT command, you must have all the authorities being copied. If you do not have an authority, that authority is not granted to the target profile. The system issues a message for each authority that is granted or not granted to the target user profile. Print the job log for a complete record. To avoid having a partial set of authorities copied, the GRTUSRAUT command should be run by a user with *ALLOBJ special authority.

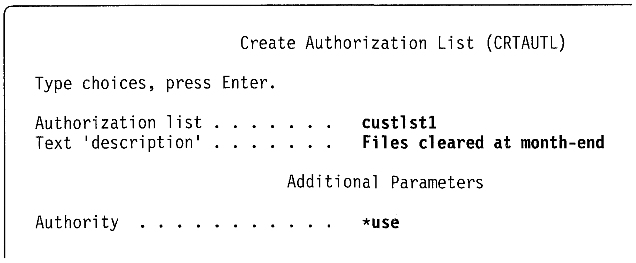
Working with Authorization Lists

Setting up an authorization list requires three steps:

1. Creating the authorization list.
2. Adding users to the authorization list.
3. Securing objects with the authorization list.

Steps 2 and 3 can be done in any order.

Creating an Authorization List: To create an authorization list, you must have *ADD authority to the QSYS library. Use the Create Authorization List (CRTAUTL) command:

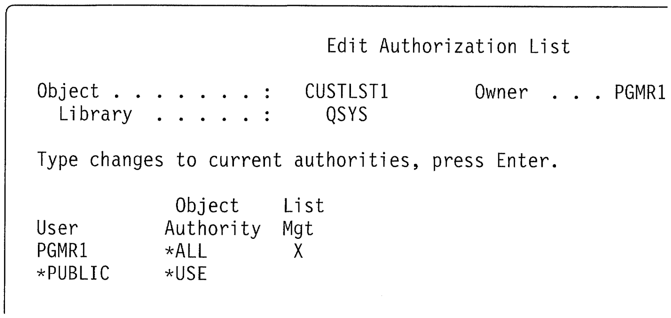


```
                  Create Authorization List (CRTAUTL)

Type choices, press Enter.

Authorization list . . . . . . .   custlst1
Text 'description' . . . . . . .   Files cleared at month-end

                        Additional Parameters

Authority  . . . . . . . . . . .   *use
```

The AUT parameter sets the public authority for any objects secured by the list. The public authority from the authorization list is used only when the public authority for an object secured by the list is *AUTL.

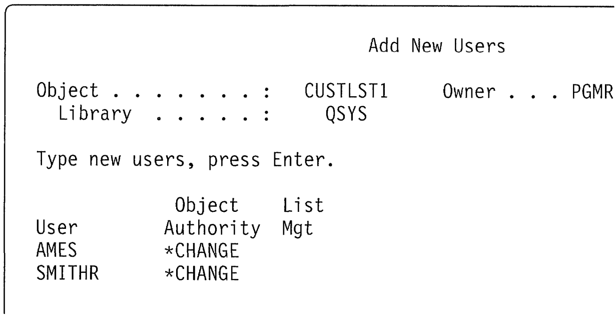
Giving Users Authority to an Authorization List: To work with the authority that user's have to the authorization list, you must have *AUTLMGT (authorization list management) authority, as well as the specific authorities you are granting. See the topic "Authorization List Management" on page 5-4 for a complete description.

You can use the Edit Authorization List (EDTAUTL) display to change user authority to the authorization list or to add new users to the list:



```
                      Edit Authorization List

Object . . . . . . . :   CUSTLST1         Owner  . . . PGMR1
  Library  . . . . . :     QSYS

Type changes to current authorities, press Enter.

           Object    List
User       Authority Mgt
PGMR1      *ALL       X
*PUBLIC    *USE
```

To give new users authority to the authorization list, press F6 (Add new users):



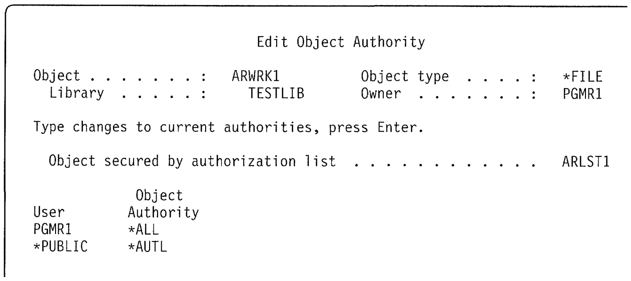
```
                         Add New Users

Object . . . . . . . :   CUSTLST1      Owner . . . PGMR1
  Library  . . . . . :     QSYS

Type new users, press Enter.

           Object    List
User       Authority Mgt
AMES       *CHANGE
SMITHR     *CHANGE
```

Each user's authority to the list is actually stored as a private authority in that user's profile. You can also use commands to work with authorization list users, either interactively or in batch:

- Add Authorization List Entry (ADDAUTLE) to define authority for additional users
- Change Authorization List Entry (CHGAUTLE) to change authority for users who are already authorized to the list
- Remove Authorization List Entry (RMVAUTLE) to remove a user's authority to the list.

Securing Objects with an Authorization List: To secure an object with an authorization list, you must own the object, have *ALL authority to it, or have *ALLOBJ special authority. You must not have *EXCLUDE authority to the authorization list.

Use the Edit Object Authority display or the GRTOBJAUT command to secure an object with an authorization list:



```
                        Edit Object Authority

Object . . . . . . . :   ARWRK1        Object type  . . . . :   *FILE
  Library  . . . . . :     TESTLIB     Owner  . . . . . . . :   PGMR1

Type changes to current authorities, press Enter.

  Object secured by authorization list  . . . . . . . . . . . .   ARLST1

           Object
User       Authority
PGMR1      *ALL
*PUBLIC    *AUTL
```

Set the public authority for the object to *AUTL if you want public authority to come from the authorization list.

On the Edit Authorization List display, you can use F15 (Display auth list objects) to list all the objects secured by the list. This is an information list only. You cannot add or remove objects from the list. You can also use the Display Authorization List Objects (DSPAUTLOBJ) command to view or print a list of all objects secured by the list.

Deleting an Authorization List: You cannot delete an authorization list if it is used to secure any objects. Use the DSPAUTLOBJ command to list all the objects secured by the list. Use either the Edit Object Authority display or the Revoke Object Authority (RVKOBJAUT) command to change the authority for each object. When the authorization list no longer secures any objects, use the Delete Authorization List (DLTAUTL) command to delete it.

Chapter 6. Security and Work Management

This chapter discusses security issues associated with work management on the system:

Job initiation
Workstations
Subsystem descriptions
Job descriptions
Library lists
Printing
Network attributes
Performance tuning

For complete information about work management topics, see the *Work Management Guide*.

Security and Job Initiation

When you start a job on the system, objects are associated with the job, such as an output queue, a job description, and the libraries on the library list. Authority for some of these objects is checked before the job is allowed to start and for other objects after the job starts. Inadequate authority may cause errors or may cause the job to end.

Objects that are part of the job structure for a job may be specified in the job description, the user profile, and on the Submit Job (SBMJOB) command for a batch job.

Starting an Interactive Job

Following is a description of the security activity performed when an interactive job is started. Because many possibilities exist for specifying the objects used by a job, this is only an example.

When an authority failure occurs during the sign-on process, a message appears at the bottom of the Sign On display describing the error. Some authority failures also cause a job log to be written. If a user is unable to sign on because of an authority failure, either change the user's profile to specify a different object or grant the user authority to the object.

After the user enters a user ID and password, these steps are performed before a job is actually started on the system:

1. The user profile and password are verified. The status of the user profile must be *ENABLED.
2. The user's authority to use the workstation is checked. See "Security and Workstations" on page 6-2 for details.
3. The system verifies authority for the values in the user profile and in the user's job description that are used to build the job structure, such as:

   Job description
   Output queue
   Current library
   Libraries in library list

   If inadequate authority exists for any of these objects, a message is displayed at the bottom of the Sign On display, and the user is unable to sign on. If authority is successfully verified for these objects, the job is started on the system.

   Note: Authority to the print device and job queue is not verified until the user attempts to use them.

After the job is started, these steps are performed before the user sees the first display or menu:

1. If the routing entry for the job specifies a user program, normal authority checking is done for the program, the program library, and any objects used by the program. If authority is not adequate, a message is sent to the user on the Sign On display and the job ends.
2. If the routing entry specifies the command processor (QCMD):
   a. Authority checking is done for the QCMD processor program, the program library, and any objects used, as described in step 1.
   b. The user's authority to the Attention-key-handling program and library is checked. If authority is not adequate, a message is sent to the user and written to the job log. Processing continues.

      If authority is adequate, the Attention-key-handling program is activated. The program is not started until the first time the user presses the Attention key. At that time, normal authority checking is done for the objects used by the program.
   c. Normal authority checking is done for the initial program (and its associated objects) specified in the user profile. If authority is adequate, the program is started. If authority is not adequate, a message is sent to the user and written to the job log. The job ends.
   d. Normal authority checking is done for the initial menu (and its associated objects) specified in the user profile. If authority is adequate, the menu is displayed. If authority is not adequate, a message is sent to the user and written to the job log. The job ends.

Starting a Batch Job

Following is a description of the security activity performed when a batch job is started. Because several methods exist for submitting batch jobs and for specifying the objects used by the job, this is only a guideline. This example uses a job submitted from an interactive job using the submit job (SBMJOB) command.

When you enter the SBMJOB command, this checking is performed before the job is added to the job queue:

1. If you specify a user profile on the SBMJOB command, you must have *USE authority to the user profile.
2. Authority is checked for objects specified as parameters on the SBMJOB command and in the job description. Authority is checked for the user profile the job will run under.
3. If the security level is 40 and the SBMJOB command specifies USER(*JOBD), the user submitting the job must have *USE authority to the user profile in the job description.
4. If authority is not adequate, a message is sent to the user and the job is not submitted.

When the system selects the job from the job queue and attempts to start the job, the authority checking sequence is similar to the sequence for starting an interactive job.

Adopted Authority and Batch Jobs

When a new job is started, a new program stack is created for the job. Adopted authority cannot take effect until the first program is added to the program stack. Adopted authority cannot be used to gain access to any objects, such as an output queue or a job description, that are added to the job structure before the job is routed. Therefore, even if your interactive job is running under adopted authority when you submit a job, that adopted authority is not used when authority is checked for the objects on your SBMJOB request.

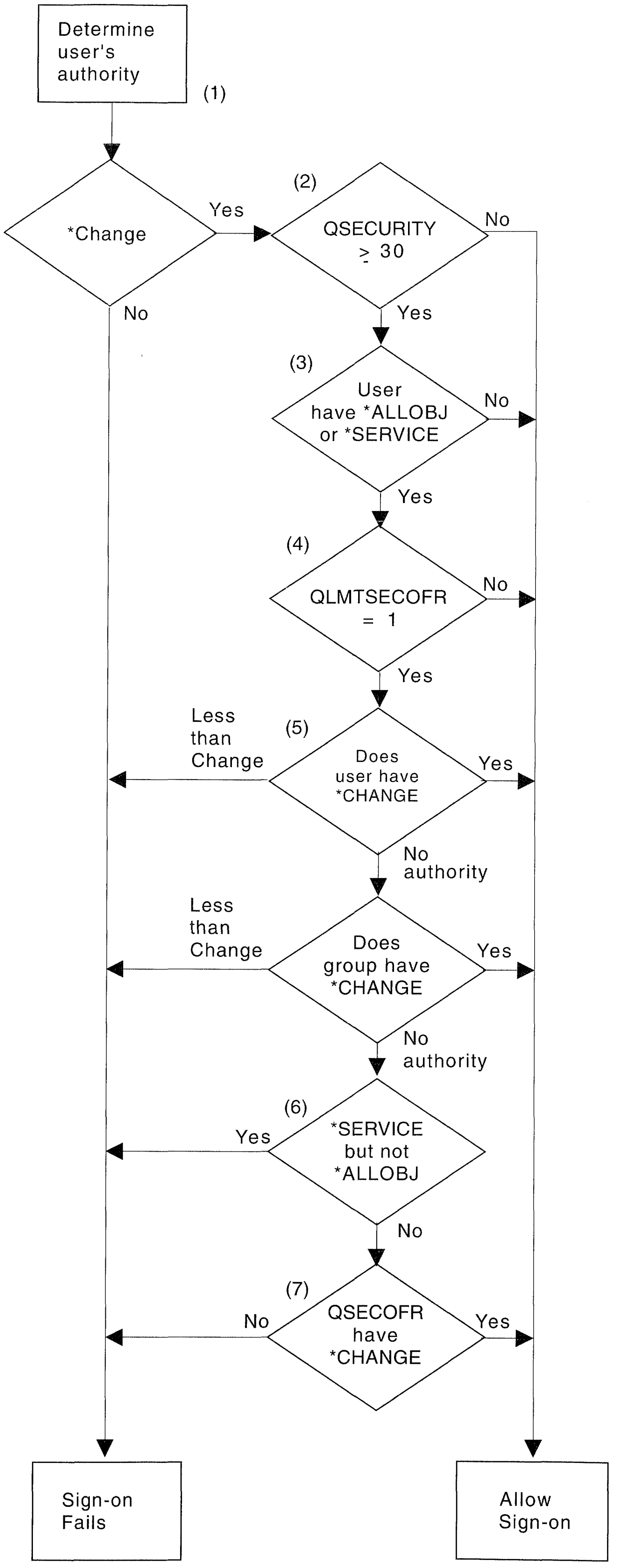
You can change characteristics of a batch job when it is waiting to run, using the Change Job (CHGJOB) command. The authority of the user under which the job will run is checked for any objects specified on the CHGJOB command. Any adopted authority of the user changing a batch job does not apply.

Security and Workstations

A **device description** contains information about a particular device or logical unit that is attached to the system. When you sign on the system, your workstation is attached to either a physical or virtual device description. To successfully sign on, you must have *CHANGE authority to the device description.

The QLMTSECOFR (limit security officer) system value controls whether users with *ALLOBJ or *SERVICE special authority must be specifically authorized to device descriptions.

Figure 6-1 shows the logic for determining whether a user is allowed to sign on at a device:



RV2L248-0

Figure 6-1. Authority Checking for Display Stations

1 Normal authority checking is performed to determine whether the user has at least *CHANGE authority to the device description. *CHANGE authority may be found using *ALLOBJ special authority from the user or group profile, private authority to the device description in the user or group profile, authority to an authorization list used to secure the device description, or public authority. Adopted authority does not apply, because authority checking for the device description is done before any programs are in the program stack for the job.

Note: The security officer (QSECOFR), service (QSRV), and basic service (QSRVBAS) user profiles are always allowed to sign on at the console. The QCONSOLE (console) system value is used to determine which device is the console. If one of these profiles attempts to sign on at the console and does not have *CHANGE authority, the system grants *CHANGE authority to the profile and allows sign-on.

2 If the security level (QSECURITY) system value on the system is less than 30, the limit security officer (QLMTSECOFR) system value is not enforced. Any user with *CHANGE authority is allowed to sign on to the device.

3 If the user profile attempting to sign on does not have *ALLOBJ or *SERVICE special authority, no additional checking is done. The user is allowed to sign on.

4 If the limit security (QLMTSECOFR) system value is 0 (No), no additional checking is done. The user is allowed to sign on. The intent of the QLMTSECOFR system value is to control which workstations certain powerful user profiles are allowed to use. It may be used to prevent users with *ALLOBJ or *SERVICE special authority from using remote workstations, dial-in workstations, or workstations located in private locations.

5 If the QLMTSECOFR system value is 1 (Yes), any user with *ALLOBJ or *SERVICE authority must have specific authority to the device. The user's profile is checked for specific authority first. If the user's profile does not have specific authority, the user's group profile is checked.

6 If the user has *SERVICE special authority, but not *ALLOBJ special authority, and no specific authority has been found, the user is not allowed to sign on at the workstation.

7 If the QSECOFR profile has specific authority to the workstation, a user with *ALLOBJ special authority is allowed to sign on.

Ownership of Device Descriptions

The default public authority on the CRTDEVxxx commands is *LIBCRTAUT. Devices are created in library QSYS, which is shipped with a CRTAUT value of *SYSVAL. The shipped value for the QCRTAUT system value is *CHANGE.

To limit the users who can sign on at a workstation, set the public authority for the workstation to *EXCLUDE and give *CHANGE authority to specific users or groups.

The security officer (QSECOFR) is not specifically given authority to any devices. If the QLMTSECOFR system value is set to 1 (YES), you must give the security officer *CHANGE authority to devices. Anyone with *OBJMGT and *CHANGE authority to a device can give *CHANGE authority to another user.

If a device description is created by the security officer, the security officer owns that device and is specifically given *ALL authority to it. When the system automatically configures devices, those devices are owned by the QPGMR profile. If you plan to use the QLMTSECOFR system value to limit where the security officer can sign on, any devices you create should be owned by a profile other than QSECOFR.

To change ownership of a display device description, the device must be powered on and varied on. Sign on at the device and change the ownership using the CHGOBJOWN command. If you are not signed on at the device, you must allocate the device before changing ownership, using the Allocate Object (ALCOBJ) command. You can allocate the device only if no one is using it. After you have changed ownership, deallocate the device using the Deallocate Object (DLCOBJ) command.

Security and Subsystem Descriptions

Subsystem descriptions control:

How jobs enter your system
How jobs are started
Performance characteristics of jobs

Only a few users should be authorized to change subsystem descriptions, and changes should be carefully monitored.

Controlling How Jobs Enter the System

Several subsystem descriptions are shipped with your system. After you have changed your security level (QSECURITY system value) to level 20 or higher, signing on without entering a user ID and password is not allowed with the subsystems shipped by IBM.

However, defining a subsystem description and job description combination that allows default sign-on (no user ID and password) is possible and represents a security exposure. When the system routes an interactive job, it looks at the workstation entry in the subsystem description for a job description. If the job description specifies USER(*RQD), the user must enter a valid user ID (and password) on the Sign On display. If the job description specifies a user profile in the *User* field, anyone can press the Enter key to sign on as that user.

| At security level 40 and higher, the system does not permit default sign-on, even if a combination of workstation entry and job description exists that would allow it. At security levels 30 and higher, the system logs an entry (type AF, subtype S) in the audit journal, if default sign-on is attempted

and the auditing function is active. See "Signing On without Password" on page 2-5 for more information.

Make sure all workstation entries for interactive subsystems refer to job descriptions with USER(*RQD). Control the authority to change job descriptions and monitor any changes that are made to job descriptions. If the auditing function is active, the system writes a JD type journal entry every time the USER parameter in a job description is changed.

Communications entries in a subsystem description control how communications jobs enter your system. A communications entry points to a default user profile, which allows a job to be started without a user ID and password. This represents a potential security exposure. Evaluate the communications entries on your system and use network attributes to control how communications jobs enter your system.

Security and Job Descriptions

A job description is a valuable tool for security and work management. You can set up a job description for a group of users who need the same initial library list, output queue, and job queue. You can set up a job description for a group of batch jobs that have similar requirements.

A job description also represents a potential security exposure. In some cases, a job description that specifies a profile name for the USER parameter can allow a job to enter the system without appropriate security checking. "Controlling How Jobs Enter the System" on page 6-3 discusses how this can be prevented for interactive and communications jobs.

When a batch job is submitted, the job might run using a different profile other than the user who submitted the job. The profile can be specified on the SBMJOB command, or it can come from the USER parameter of the job description. If your system is at security level (QSECURITY system value) 30 or lower, the user submitting a job needs authority to the job description but not to the user profile specified on the job description. This represents a security exposure. At security level 40 and higher, the submitter needs authority to both the job description and the user profile.

For example:

- USERA is not authorized to file PAYROLL.
- USERB has *USE authority to the PAYROLL file and to program PRLIST, which lists the PAYROLL file.
- Job description PRJOBD specifies USER(USERB). Public authority for PRJOBD is *USE.

At security level 30 or lower, USERA can list the payroll file by submitting a batch job:

```
SBMJOB RQSDTA('Call PRLIST') JOBD(PRJOBD) +
       USER(*JOBD)
```

You can prevent this by using security level 40 and higher or by controlling the authority to job descriptions that specify a user profile. The Check Job Description User (CHKJOBDUSR) tool in the QUSRTOOL library can assist you in monitoring job descriptions that specify user profile names.

Sometimes, a specific user profile name in a job description is required for certain types of batch work to function properly. For example, the QBATCH job description is shipped with USER(QPGMR). This job description is shipped with the public authority of *CHANGE.

If your system is at security level 30 or lower, any user on the system who has authority to the Submit Job (SBMJOB) command or the start reader commands can submit work under the programmer (QPGMR) user profile, whether or not the user has authority to the QPGMR profile. At security level 40 and higher, *USE authority to the QPGMR profile is required. Depending on your security needs, you may want to change the public authority of the QBATCH job description to *EXCLUDE,

Security and the System Operator Message Queue

The message handling option from the Operational Assistant menu (ASSIST) provides a function key to work with system operator messages. You may want to prevent users from responding to messages in the QSYSOPR (system operator) message queue. Incorrect responses to system operator messages can cause problems on your system.

Responding to messages requires *USE and *ADD authorities to the message queue. Removing messages requires *USE and *DLT authorities. (See page D-41.) Give the authority to respond to and remove messages in QSYSOPR only to users with system operator responsibility. Public authority to QSYSOPR should be *OBJOPR and *ADD, which allows adding new messages to QSYSOPR.

Warning: All jobs need the ability to add new messages to the QSYSOPR message queue. Do not make the public authority to QSYSOPR *EXCLUDE.

Security and Library Lists

The **library list** for a job indicates which libraries are to be searched and the order in which they are to be searched. When a program specifies an object, the object can be specified with a qualified name, which includes both the object name and the library name. Or, the library for the object can be specified as *LIBL (library list). The libraries on the library list are searched, in order, until the object is found.

Table 6-1 on page 6-5 summarizes the parts of the library list and how they are built during a job. The sections that follow discuss the risks and protection measures for library lists.

Table 6-1. Parts of the Library List. The library list is searched in this sequence:

Part	How It Is Built
System Portion 15 entries	Initially built using the QSYSLIBL system value. Can be changed during a job with the CHGSYSLIBL command.
Product Library Portion 2 entries	Initially blank. A library is added to the product library portion of the library list when a command or menu runs that was created with a library in the PRDLIB parameter. The library remains in the product library portion of the library list until the command or menu ends.
Current Library 1 entry	Specified in the user profile or on the Sign On display. Can be changed when a command or menu runs that specifies a library for the CURLIB parameter. Can be changed during the job with the CHGCURLIB command.
User Portion 25 entries	Initially built using the initial library list from the user's job description. If the job description specifies *SYSVAL, the QUSRLIBL system value is used. During a job, the user portion of the library list can be changed with the ADDLIBLE, RMVLIBLE, CHGLIBL, and EDTLIBL commands.

Security Risks of Library Lists

Library lists represent a potential security exposure. If a user is able to change the sequence of libraries on the library list, or add additional libraries to the list, the user may be able to perform functions that break your security requirements.

Following are two examples of how changes to a library list might break security requirements:

Change in Function: Figure 6-2 shows an application library. Program A calls Program B, which is expected to be in LIBA. Program B performs updates to File A. Program B is called without a qualified name, so the library list is searched until Program B is found.

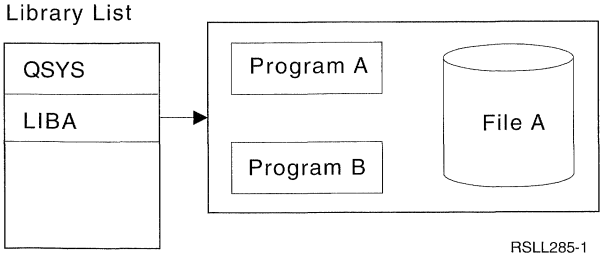


Figure 6-2. Expected Environment

A programmer or another knowledgeable user could place another Program B in the library LIBB. The substitute program might perform different functions, such as making a copy of confidential information or updating files incorrectly.

If LIBB is placed ahead of LIBA in the library list, the substitute Program B is run instead of the original Program B, because the program is called without a qualified name:

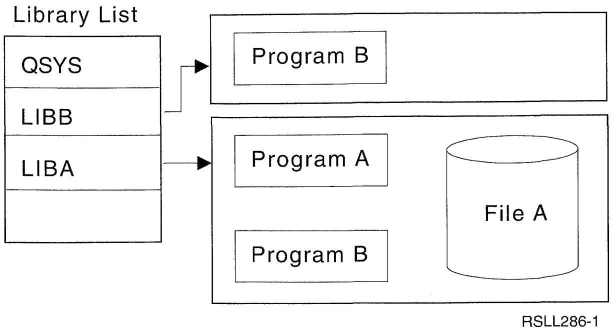


Figure 6-3. Actual Environment

Unauthorized Access to Information: Assume Program A in Figure 6-2 adopts the authority of USER1, who has *ALL authority to File A. Assume Program B is called by Program A (adopted authority remains in effect). A knowledgeable user could create a substitute Program B which simply calls the command processor. The user would have a command line and complete access to File A.

Recommendations for System Portion of Library List

The system portion of the library list is intended for IBM-supplied libraries. Application libraries that are carefully controlled can also be placed in the system portion of the library list. The system portion of the library list represents the greatest security exposure, because the libraries in this part of the list are searched first.

Only a user with *ALLOBJ and *SECADM special authority can change the QSYSLIBL system value. Control and monitor any changes to the system portion of the library list. Follow these guidelines when adding libraries:

- Only libraries that are specifically controlled should be placed on this list.
- The public should not have *ADD authority to these libraries.
- A few IBM-supplied libraries, such as QGPL, QRJE and QUSRSYS, are shipped with public authority *ADD for production reasons. Regularly monitor what objects, particularly programs and commands, are added to these libraries.

The CHGSYSLIBL command is shipped with public authority *EXCLUDE. Only users with *ALLOBJ authority are authorized to the command, unless you grant authority to other users. If the system library list needs to be changed temporarily during a job, you can use the technique described in the topic "Changing the System Library List" on page 7-3.

Recommendations for Product Library

The product library portion of the library list is searched before the user portion. A knowledgeable user could create a command or menu that inserts a product library into the library list. For example, this statement creates CMDX, which runs program PGMA:

```
CRTCMD CMDX PGM(PGMA) PRDLIB(LIBB)
```

As long as PGMA is running, LIBB is in the product portion of the library list.

Use these measures to protect the product portion of the library list:

- Control authority to the Create Command (CRTCMD), Change Command (CHGCMD), Create Menu (CRTMNU), and Change Menu (CHGMNU) commands.
- When you create commands and menus, specify PRDLIB(*NONE), which removes any entries currently in the product portion of the library list. This protects you from having unknown libraries searched ahead of the library you expect when your command or menu runs.

  Note: The default when you create a command or menu is PRDLIB(*NOCHG). *NOCHG means that when the command or menu is run, the product library portion of the library list is not changed.

Recommendations for the Current Library

The current library can be used by decision-support tools, such as Query/400. Any query programs created by a user are, by default, placed in the user's current library. When you create a menu or command, you can specify a current library to be used while the menu is active.

The current library provides an easy method for the user and the programmer to create new objects, such as query programs, without worrying about where they should be located. However, the current library poses a security risk, because it is searched before the user portion of the library list. You can take several precautions to protect the security of your system while still making use of the current library capability:

- Specify *YES for the *Limit capabilities* field in the user profile. This prevents a user from changing the current library on the Sign On display or using the CHGPRF command.
- Restrict authority to the Change Current Library (CHGCURLIB), CRTMNU, CHGMNU, CRTCMD, and CHGCMD commands.
- Use the technique described in "Controlling the User Library List" on page 7-3 to set the current library during application processing.

Recommendations for the User Portion of the Library List

The user portion of the library list usually changes more than the other portions and is more difficult to control. Many application programs change the library list. Job descriptions also affect the library list for a job.

Following are some suggested alternatives for controlling the user portion of the library list to make sure unauthorized libraries with substitute programs and files are not used during processing:

- Restrict users of production applications to a menu environment. Set the *Limit capabilities* field in user profiles to *YES to restrict their ability to enter commands. "Planning Menus" on page 7-4 provides an example of this environment.
- Use qualified names (object and library) in your applications. This prevents the system from searching the library list to find an object.
- Control the ability to change job descriptions, because the job description sets the initial library list for a job.
- Use the Add Library List Entry (ADDLIBLE) command at the beginning of the program to ensure the desired objects are at the beginning of the user portion of the library list. At the end of the program, the library can be removed.

  If the library is already on the library list, but you are not sure if it is at the beginning of the list, you must remove the library and add it. If the sequence of the library list is important to other applications on the system, use the next method instead.
- Use a program that retrieves and saves the library list for a job. Replace the library list with the list desired for the application. When the application ends, return the library list to its original setting. See "Controlling the User Library List" on page 7-3 for an example of this technique.

Security and Printing

Most information that is printed on your system is stored as a spooled file on an output queue while it is waiting to print. Unless you control the security of output queues on your system, unauthorized users can display, print, and even copy confidential information that is waiting to print.

One method for protecting confidential output is to create a special output queue. Send confidential output to the output queue and control who can view and manipulate the spooled files on the output queue.

To determine where output goes, the system looks at the printer file, job attributes, user profile, workstation device description, and the print device (QPRTDEV) system value in sequence. If defaults are used, the output queue associated with the QPRTDEV printer is used. The *Guide to Program-*

ming for Printing provides examples of how to direct output to a particular output queue.

Securing Spooled Files

A spooled file is a special type of object on the system. You cannot directly grant and revoke authority to view and manipulate a spooled file. The authority to a spooled file is controlled by several parameters on the output queue that holds the spooled file.

When you create a spooled file, you are the owner of that file. You can always view and manipulate any spooled files you own, regardless of how the authority for the output queue is defined. You must have *READ authority to add new entries to an output queue. If your authority to an output queue is removed, you can still access any entries you own on that queue using the Work with Spooled Files (WRKSPLF) command.

The security parameters for an output queue are specified using the Create Output Queue (CRTOUTQ) command or the Change Output Queue (CHGOUTQ) command. You can display the security parameters for an output queue using the Work with Output Description (WRKOUTQD) command.

Warning: A user with *SPLCTL special authority can perform all functions on all entries, regardless of how the output queue is defined.

Display Data (DSPDTA) Parameter of Output Queue: The DSPDTA parameter is designed to protect the contents of a spooled file. It determines what authority is required to perform the following functions on spooled files owned by other users:

- View the contents of a spooled file (DSPSPLF command)
- Copy a spooled file (CPYSPLF command)
- Send a spooled file (SNDNETSPLF command)
- Move a spooled file to another output queue (CHGSPLFA command)

Possible Values for DSPDTA

<u>*NO</u> A user cannot display, send, or copy spooled files owned by other users, unless the user has one of the following:

- *JOBCTL special authority if the OPRCTL parameter is *YES.
- *CHANGE authority to the output queue if the *AUTCHK parameter is *DTAAUT.
- Ownership of the output queue if the *AUTCHK parameter is *OWNER.

***YES** Any user with *READ authority to the output queue can display, copy, or send the data of spooled files owned by others.

Possible Values for DSPDTA

***OWNER** Only the owner of a spooled file can display, copy, send, or move the file. If the OPRCTL value is *YES, users with *JOBCTL special authority can hold, change, delete, and release spooled files on the output queue, but they cannot display, copy, send, or move the spooled files. This is intended to allow operators to manage entries on an output queue without being able to view the contents.

Authority to Check (AUTCHK) Parameter of Output Queue: The AUTCHK parameter determines whether *CHANGE authority to the output queue allows a user to change and delete spooled files owned by other users.

Possible Values for AUTCHK

<u>*OWNER</u> Only the user who owns the output queue can change or delete spooled files owned by others.

***DTAAUT** Specifies that any user with *READ, *ADD, and *DLT authority to the output queue can change or delete spooled files owned by others.

Operator Control (OPRCTL) Parameter of Output Queue: The OPRCTL parameter determines whether a user with *JOBCTL special authority can control the output queue.

Possible Values for OPRCTL

<u>*YES</u> A user with *JOBCTL special authority can perform all functions on the spooled files, unless the DSPDTA value is *OWNER. If the DSPDTA value is *OWNER, *JOBCTL special authority does not allow the user to display, copy, send, or move spooled files.

***NO** *JOBCTL special authority does not give the user any authority to perform operations on the output queue. Normal authority rules apply to the user.

Authority Required to Use Printing Functions

Table 6-2 on page 6-8 shows what combination of output queue parameters and authority to the output queue is required to perform print management functions on the system. For some functions, more than one combination is listed. The owner of a spooled file can always perform all functions on that file.

The authority and output queue parameters for all commands associated with spooled files are shown on page D-58. Output queue commands are shown on page D-46.

Warning: A user with *SPLCTL (spool control) special authority is not subject to any authority restrictions associated with output queues. *SPLCTL special authority allows the user to perform all operations on all output queues. Carefully evaluate giving *SPLCTL special authority to any user.

Table 6-2. Authority Required to Perform Printing Functions

	Output Queue Parameters			Output Queue Authority	Special Authority
Printing Function	**DSPDTA**	**AUTCHK**	**OPRCTL**		
Add spooled files to queue [1]	Any	Any	Any	*READ	None
	Any	Any	*YES	Any	*JOBCTL
View list of spooled files	Any	Any	Any	*READ	None
(WRKOUTQ command [2])	Any	Any	*YES	Any	*JOBCTL
Display, copy, or send spooled files	*YES	Any	Any	*READ	None
(DSPSPLF, CPYSPLF, SNDNETSPLF, SNDTCPSPLF [2])	*NO	*DTAAUT	Any	*CHANGE	None
	*NO	*OWNER	Any	Owner [3]	None
	*YES	Any	*YES	Any	*JOBCTL
	*NO	Any	*YES	Any	*JOBCTL
	*OWNER [5]	Any	Any	Any	Any
Change, delete, hold, and release spooled file	Any	*DTAAUT	Any	*CHANGE	None
(CHGSPLFA, DLTSPLF, HLDSPLF, RLSSPLF [2])	Any	*OWNER	Any	Owner [3]	None
	Any	Any	*YES	Any	*JOBCTL
Change, clear, hold, and release output queue	Any	*DTAAUT	Any	*CHANGE	None
(CHGOUTQ, CLROUTQ, HLDOUTQ, RLSOUTQ [2])	Any	*OWNER	Any	Owner [3]	None
	Any	Any	*YES	Any	*JOBCTL
Start a writer for the queue	Any	Any	Any	*CHANGE [4]	None
(STRPRTWTR [2])	Any	Any	*YES	Any [4]	*JOBCTL

[1] This is the authority required to direct your output to an output queue.

[2] Using these commands or equivalent options from a display.

[3] You must be the owner of the output queue.

[4] Also requires *USE authority to the printer device description.

[5] You must be the owner of the spooled file or have *SPLCTL special authority.

Output Queue Examples

Following are several examples of setting security parameters for output queues to meet different requirements:

- Create a general-purpose output queue. All users are allowed to display all spooled files. The system operators are allowed to manage the queue and change spooled files:

```
CRTOUTQ OUTQ(QGPL/GPOUTQ) DSPDTA(*YES) +
        OPRCTL(*YES) AUT(*USE)
```

- Create an output queue for an application. Only members of the group profile GRPA are allowed to use the output queue. All authorized users of the output queue are allowed to display all spooled files. System operators are not allowed to work with the output queue:

```
CRTOUTQ OUTQ(ARLIB/AROUTQ) DSPDTA(*YES) +
        OPRCTL(*NO) AUT(*EXCLUDE)
GRTOBJAUT OBJ(ARLIB/AROUTQ) OBJTYP(*OUTQ) +
        USER(GRPA)  AUT(*CHANGE)
```

- Create a confidential output queue for the security officers to use when printing information about user profiles and authorities. The output queue is created and owned by the QSECOFR profile.

```
CRTOUTQ OUTQ(QGPL/SECOUTQ) DSPDTA(*OWNER) +
        AUTCHK(*DTAAUT) OPRCTL(*NO) +
        AUT(*EXCLUDE)
```

  Even if the security officers on a system have *ALLOBJ special authority, they are not able to access spooled files owned by others on the SECOUTQ output queue.

- Create an output queue that is shared by users printing confidential files and documents. Users can work with only their own spooled files. System operators can work with the spooled files, but they cannot display the contents of the files.

```
CRTOUTQ OUTQ(QGPL/CFOUTQ) DSPDTA(*OWNER) +
        AUTCHK(*OWNER) OPRCTL(*YES) AUT(*USE)
```

Security and Network Attributes

Network attributes control how your system communicates with other systems. Some network attributes control how remote requests to process jobs and access information are handled. These network attributes directly affect security on your system and are discussed in the sections that follow:

Job action (JOBACN)
PC Support access (PCSACC)
Distributed data management (DDMACC)

Possible values for each network attribute are shown. The default value is underlined. To set the value of a network attribute, use the Change Network Attribute (CHGNETA) command.

Job Action (JOBACN) Network Attribute

The JOBACN network attribute determines how the system processes incoming requests to run jobs.

Possible Values for JOBACN:

***REJECT** The input stream is rejected. A message stating the input stream was rejected is sent to both the sender and the intended receiver.

***FILE** The input stream is filed on the queue of network files for the receiving user. This user can display, cancel, or receive the input stream into a database file or submit it to a job queue. A message stating that the input stream was filed is sent to both the sender and the receiver.

***SEARCH** The network job table controls the actions by using the values in the table.

Recommendations: If you do not expect to receive remote job requests on your system, set the JOBACN network attribute to *REJECT.

For more information about the JOBACN attribute, refer to the *Distribution Services Network Guide*.

PC Support Access (PCSACC) Network Attribute

The PCSACC network attribute determines how the PC Support/400 licensed program processes requests from attached personal computers to access objects. The PCSACC network attribute controls whether personal computer jobs can access objects on the AS/400 system, not whether the personal computer can use workstation emulation.

Possible Values for PCSACC:

***REJECT** PC Support rejects every request from the personal computer to access objects on the AS/400 system. An error message is sent to the PC application.

***OBJAUT** The PC Support programs on the system verify normal object authorities for any object requested by a PC program. For example, if file transfer is requested, authority to copy data from the database file is checked.

qualified-program-name The PC Support program calls this user-written exit program to determine if the PC request should be rejected. The exit program is called only if normal authority checking for the object is successful. The PC Support program passes information about the user and the requested function to the exit program. The program returns a code indicating whether the request should be allowed or rejected. If the return code indicates the request should be rejected or if an error occurs, an error message is sent to the personal computer.

Risks and Recommendations: Normal security measures on your system may not be sufficient protections if the PC Support/400 program is installed on your system. For example, if a user has *USE authority to a file and the PCSACC network attribute is *OBJAUT, the user can use the PC Support program and a program on the personal computer to transfer that entire file to the personal computer. The user can then copy the data to a PC diskette or tape and remove it from the premises.

Several methods are available to prevent an AS/400 workstation user with *USE authority to a file from copying the file:

- Setting LMTCPB(*YES) in the user profile.
- Restricting authority to commands that copy files.
- Not giving the user *ADD authority to any library. *ADD authority is required to create a new file in a library.
- Not giving the user access to any *SAVRST device.

None of these methods work for the PC user of the PC Support/400 licensed program. Using an exit program to verify all requests is the only adequate protection measure.

The PC Support program passes information for the following types of access to the user exit program called by the PCSACC network attribute:

File transfer
Virtual print
Message
Shared folder

The *PC Support/400 Technical Reference for DOS and OS/2* manual contains a complete description of the information the system passes to the user-written exit program.

Distributed Data Management Access (DDMACC) Network Attribute

The DDMACC network attribute determines how the system processes requests from other systems to access data using the distributed data management (DDM) or the distributed relational database function.

Possible Values for DDMACC:

***REJECT** The system does not allow any DDM requests from remote systems. *REJECT does not prevent this system from functioning as the requester system and sending requests to other server systems.

***OBJAUT** Remote requests are controlled by the object authority on the system.

qualified-program-name This user-written exit program is called after normal object authority has been verified. The exit program is called only for DDM files, not for distributed relational database functions. The exit program is passed a parameter list, built by the remote system, that identifies the local system user and the request. The program evaluates the request and sends a return code, granting or denying the requested access.

For more information about the DDMACC network attribute and the security issues associated with DDM, see the *DDM Guide*.

Security and Performance Tuning

Monitoring and tuning performance is not the responsibility of a security officer. However, the security officer should ensure that users are not altering the performance characteristics of the system to speed up their own jobs at the expense of others.

Several work management objects affect the performance of jobs in the system:

- The class sets the run priority and time slice for a job.
- The routing entry in the subsystem description determines the class and the storage pool the job uses.
- The job description can determine the output queue, output priority, job queue, and job priority.

Knowledgeable users with appropriate authority can create their own environment on the system and give themselves better performance than other users. Control this by limiting the authority to create and change work management objects. Set the public authority to work management commands to *EXCLUDE and grant authority to a few trusted users.

Performance characteristics of the system can also be changed interactively. For example, the Work with System Status (WRKSYSSTS) display can be used to change the size of storage pools and the activity levels. Also, a user with *JOBCTL (job control) special authority can change the scheduling priority of any job on the system, subject to the priority limit (PTYLMT) in the user's profile. Assign *JOBCTL special authority and PTYLMT in user profiles carefully.

To allow users to view performance information using the WRKSYSSTS command but not change it, do the following:

```
GRTOBJAUT OBJ(CHGSHRPOOL) OBJTYPE(*CMD) +
          USER(*PUBLIC)   AUT(*EXCLUDE)
```

Authorize users responsible for system tuning to change performance characteristics:

```
GRTOBJAUT OBJ(CHGSHRPOOL) OBJTYPE(*CMD) +
          USER(USRTUNE)   AUT(*USE)
```

Restricting Jobs to Batch

You can create or change commands to restrict certain jobs to be run only in a batch environment. For example, you may want to run certain reports or program compiles in batch. A job running in batch usually affects system performance less than the same job running interactively.

For example, to restrict the command that runs program RPTA to batch, do the following:

- Create a command to run RPTA and specify that the command can be run only in batch:

  ```
  CRTCMD CMD(RPTA) PGM(RPTA) ALLOW(*BATCH *BPGM)
  ```

To restrict compiles to batch, do the following for the create command for each program type:

```
CHGCMD CMD(CRTxxxPGM) ALLOW(*BATCH *BPGM)
```

Chapter 7. Designing Security

Protecting information is an important part of most applications. Security should be considered, along with other requirements, at the time the application is designed. For example, when deciding how to organize application information into libraries, try to balance security requirements with other considerations, such as application performance and backup and recovery.

This chapter contains guidelines to help application developers and systems managers include security as part of the overall design. It also contains examples of techniques you can use to accomplish security objectives on your system. Some of the examples in this chapter contain sample programs. These programs are included for illustrative purposes only. Many of them will not compile or run successfully as is, nor do they include message handling and error recovery.

The *Basic Security Guide* is intended for the security administrator. It contains forms, examples, and guidelines for planning security for applications that have already been developed. If you have responsibility for designing an application, you may find it useful to review the forms and examples in the *Basic Security Guide*. They can help you view your application from the perspective of a security administrator and understand what information you need to provide.

The *Basic Security Guide* uses a set of example applications for a fictional company called the JKL Toy Company. This chapter discusses design considerations for the same set of example applications. Figure 7-1 shows the relationships between user groups, applications, and libraries for the JKL Toy Company:

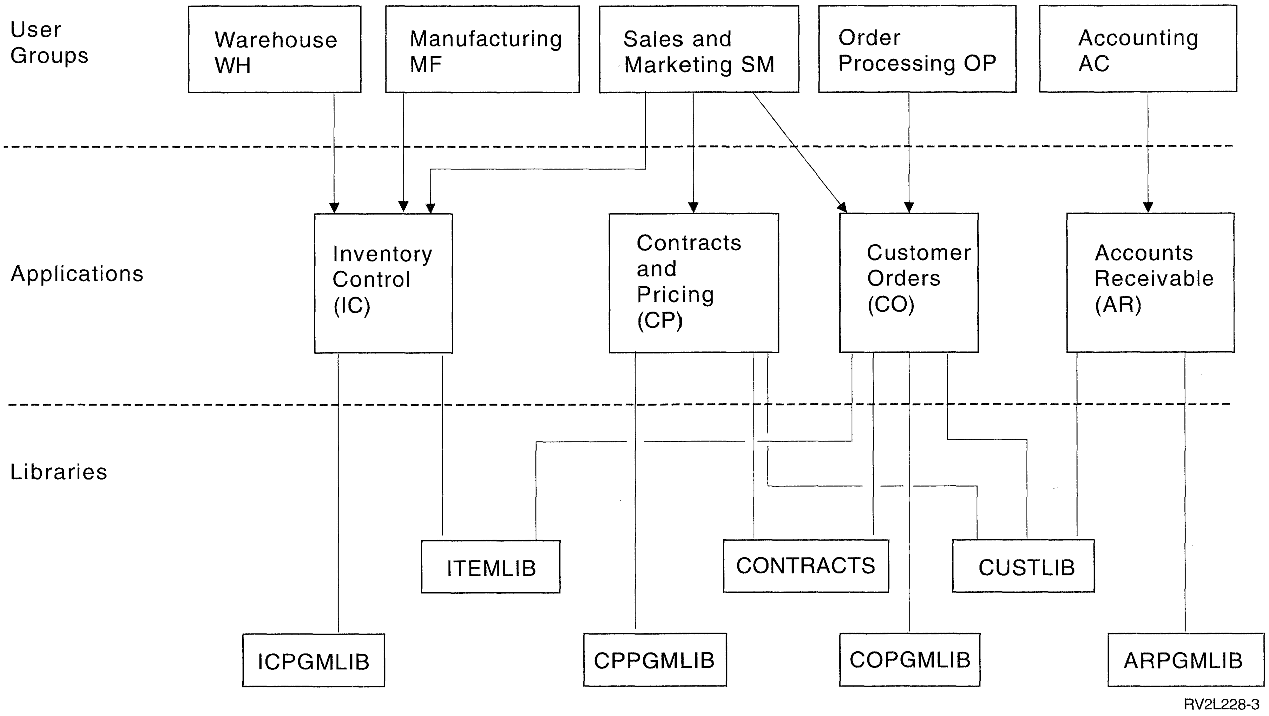


Figure 7-1. Example Applications

Overall Recommendations

The recommendations in this chapter and in the *Basic Security Guide* rely on one important principle: simplicity. Keeping your security design as simple as possible makes it easier to manage and audit security. It also improves application performance and backup performance.

Following is a list of general recommendations for security design:

- Do not rely completely on resource security. Use the other methods available, such as limited capabilities in the user profile and restricting users to a set of menus, as well as resource security, to protect information.
- Secure only those objects that really require security. Analyze a library to determine which objects, such as data files, are confidential and secure those objects. Use public authority for other objects, such as data areas and message queues.
- Move from the general to the specific:
  - Plan security for libraries. Deal with individual objects only when necessary.
  - Plan public authority first, followed by group authority and individual authority.

- Make the public authority for new objects in a library (CRTAUT parameter) the same as the public authority for the majority of existing objects in the library.
- To make auditing easier and improve authority-checking performance, avoid defining private authority that is less than the public authority for an object.
- Use authorization lists to group objects with the same security requirements. Authorization lists are simpler to manage than individual authorities and help to recover security information.
- Create special user profiles to own applications. Owner profiles that are specific to an application make it easier to recover applications and to move applications between systems. Information about private authorities is spread among several profiles, which improves performance. Owner profiles also allow you to adopt the authority of the owner profile rather than a more powerful profile that provides unnecessary authority.
- Avoid having applications owned by IBM-supplied user profiles, such as QSECOFR or QPGMR. These profiles own a large number of IBM-supplied objects and can become difficult to manage. Having applications owned by IBM-supplied user profiles can also cause security problems when moving applications from one system to another.

Planning Libraries

Many factors affect how you choose to group your application information into libraries and manage libraries. This topic addresses some of the security issues associated with library design.

To access an object, you need authority to the object itself and to the library containing the object. You can restrict access to an object by restricting the object itself, the library containing the object, or both.

A library is like a directory used to locate the objects in the library. *USE authority to a library allows you to use the directory to find objects in the library. The authority for the object itself determines *how* you can use the object. *USE authority to a library is sufficient to perform most operations on the objects in the library. See “Library Security” on page 5-3 for more information about the relationship between library and object authority.

Using public authority for objects and restricting access to libraries can be a simple, effective security technique. Putting programs in a separate library from other application objects can also simplify security planning. This is particularly true if files are shared by more than one application. You can use authority to the libraries containing application programs to control who can perform application functions.

Following are two examples of using library security for the JKL Toy Company applications. (See Figure 7-1 on page 7-1 for a diagram of the applications.)

- The information in the CONTRACTS library is considered confidential. The public authority for all the objects in the library is sufficient to perform the functions of the Pricing and Contracts application (usually *CHANGE). The public authority to the CONTRACTS library itself is *EXCLUDE. Only users or groups authorized to the Contracts and Pricing application are granted *USE authority to the library.
- The JKL Toy Company is a small company with a nonrestrictive approach to security, except for the contract and pricing information. All system users are allowed to view customer and inventory information, although only authorized users can change it. The CUSTLIB and the ITEMLIB libraries, and the objects in the libraries, have public authority of *USE. Users can view information in these libraries through their primary application or by using Query. The program libraries have public authority *EXCLUDE. Only users who are allowed to change inventory information have access to the ICPGMLIB. Programs that change inventory information adopt the authority of the application owner (OWNIC) and thus have *ALL authority to the files in the ITEMLIB library.

Library security is effective only if these rules are followed:

- Libraries contain objects with similar security requirements.
- Users are not allowed to add new objects to restricted libraries. Changes to programs in the libraries are controlled.
- Library lists are controlled.

Library Ownership

A library and the objects in the library should all be owned by the same profile. This simplifies managing security for the library, planning recovery, and moving the library to a different system. When you move a library to a different system, you or the security officer can create the owner profile on the target system. When you restore the library, ownership of the library objects is restored to the owner profile.

You can use the Check Library Owner (CHKLIBOWN) tool and the Change Library Owner (CHGLIBOWN) tool in the QUSRTOOL library to help you manage library ownership.

Library Lists

The library list for a job provides flexibility. It also represents a security exposure. This exposure is particularly important if you use public authority for objects and rely on library security as your primary means of protecting information. In this case, a user who gains access to a library has uncontrolled access to the information in the library. The topic “Security and Library Lists” on page 6-4 provides a discussion of security issues associated with library lists.

To avoid the security risks of library lists, your applications can specify qualified names. When both the object name and the library are specified, the system does not search the library list. This prevents a potential intruder from using the library list to circumvent security.

However, other application design requirements may prevent you from using qualified names. If your applications rely on library lists, the technique described in the next section can reduce the security exposure.

Controlling the User Library List: As a security precaution, you may want to make sure the user portion of the library list has the correct entries in the expected sequence before a job runs. One method for doing this is to use a CL program to save the user's library list, replace it with the desired list, and restore it at the end of the application. Following is a sample program to do this:

```
           PGM
           DCL      &USRLIBL *CHAR LEN(275)
           DCL      &CURLIB  *CHAR LEN(10)
           DCL      &ERROR *LGL
           DCL      &CMD *CHAR LEN(500)
           MONMSG   MSGID(CPF0000)  +
                    EXEC(GOTO SETERROR)
           RTVJOBA  USRLIBL(&USRLIBL) +
                    CURLIB(&CURLIB)
           IF COND(&CURLIB=*NONE) +
              THEN(CHGVAR &CURLIB '*CRTDFT   ')
           CHGLIBL  LIBL(QPGL) CURLIB(*CRTDFT)
           /*********************************/
           /*                               */
           /*     Normal processing         */
           /*                               */
           /*********************************/
           GOTO     ENDPGM
SETERROR:  CHGVAR   &ERROR '1'
ENDPGM:    CHGVAR   &CMD +
                    ('CHGLIBL LIBL+
                    (' *CAT &USRLIBL *CAT')  +
                    CURLIB(' *CAT &CURLIB *TCAT ' )')
           CALL     QCMDEXC PARM(&CMD 500)
           IF       &ERROR SNDPGMMSG MSGID(CPF9898) +
                    MSGF(QCPFMSG) MSGTYPE(*ESCAPE) +
                    MSGDTA('The xxxx error occurred')
           ENDPGM
```

Figure 7-2. Program to Replace and Restore Library List

Notes:

1. Regardless of how the program ends (normally or abnormally), the library list is returned to the version it held when the program was called, because error handling includes restoring the library list.
2. Because the CHGLIBL command requires a list of library names, it cannot be run directly. The RTVJOBA command, therefore, retrieves the libraries used to build the CHGLIBL command as a variable. The variable is passed as a parameter to the QCMDEXC function.
3. If you exit to an uncontrolled function (for example, a user program, a menu that allows commands to be entered, or the Command Entry display) in the middle of a program, your program should replace the library list on return, to ensure adequate control.

Changing the System Library List: If your application needs to add entries to the system portion of the library list, you can use a CL program similar to the one shown in Figure 7-2, with the following changes:

- Instead of using the RTVJOBA command, use the Retrieve System Values (RTVSYSVAL) command to get the value of the QSYSLIBL system value.
- Use the Change System Library List (CHGSYSLIBL) command to change the system portion of the library list to the desired value.
- At the end of your program, use the CHGSYSLIBL command again to restore the system portion of the library list to its original value.
- The CHGSYSLIBL command is shipped with public authority *EXCLUDE. To use this command in your program, do one of the following:
  - Grant the program owner *USE authority to the CHGSYSLIBL command and use adopted authority.
  - Grant users running the program *USE authority to the CHGSYSLIBL command.

Describing Library Security

As an application designer, you need to provide information about a library for the security administrator. The security administrator uses this information to decide how to secure the library and its objects. Typical information needed is:

- Any application functions which add objects to the library.
- Whether any objects in the library are deleted during application processing.
- What profile owns the library and its objects.
- Whether the library should be included on library lists.

Figure 7-3 provides a sample format for providing this information:

Library name: ITEMLIB

Public authority to the library: *EXCLUDE

Public authority to objects in the library: *CHANGE

Public authority for new objects (CRTAUT): *CHANGE

Library owner: OWNIC

Include on library lists? No. Library is added to library list by initial application program or initial query program.

List any functions that require *ADD authority to the library:

No objects are added to the library during normal application processing.

List any objects requiring *OBJMGT or *OBJEXIST authority and what functions need that authority:

All work files, whose names begin with the characters ICWRK, are cleared at month-end. This requires *OBJMGT authority.

Figure 7-3. Format for Describing Library Security

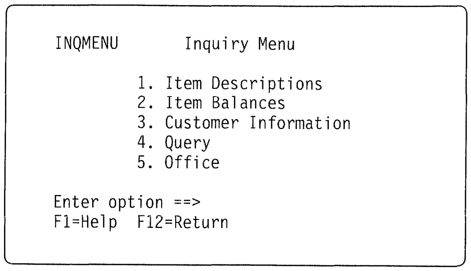
Planning Menus

Menus are a good method for providing security on your system. You can use menus to restrict a user to a set of strictly controlled functions by specifying limited capabilities and an initial menu in the user profile.

To use menus as a security tool, follow these guidelines when designing them:

- Do not provide a command line on menus designed for restricted users.
- Avoid having functions with different security requirements on the same menu. For example, if some application users are allowed to only view information, not change it, provide a menu that has only display and print options for those users.
- Make sure the set of menus provides all the necessary links between menus so the user does not need a command line to request one.
- Provide access to a few system functions, such as viewing printer output. The ASSIST system menu gives this capability and can be defined in the user profile as the Attention-key-handling program. If the user profile has a class of *USER and has limited capabilities, the user cannot view the output or jobs of other users.
- Provide access to decision-support tools and the OfficeVision/400 licensed program from menus. The topic "Using Adopted Authority in Menu Design" gives an example of how to do this.
- Consider controlling access to the System Request Menu or some of the options on this menu. See "System Request Menu" on page 7-7 for more information.
- For users who are allowed to run only a single function, avoid menus entirely and specify an initial program in the user profile. Specify *SIGNOFF as the initial menu.

At the JKL Toy Company, all users see an inquiry menu allowing access to most files. For users who are not allowed to change information, this is the initial menu. The return option on the menu signs the user off. For other users, this menu is called by an inquiry option from application menus. By pressing F12 (Return), the user returns to the calling menu. Because library security is used for program libraries, this menu and the programs it calls are kept in the QGPL library:



```
INQMENU        Inquiry Menu

            1. Item Descriptions
            2. Item Balances
            3. Customer Information
            4. Query
            5. Office

Enter option ==>
F1=Help  F12=Return
```

Figure 7-4. Sample Inquiry Menu

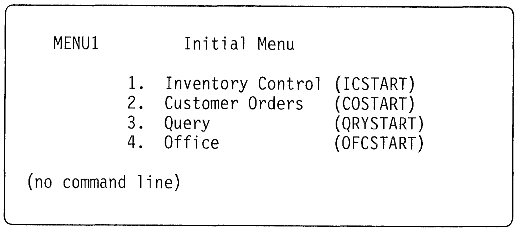
Using Adopted Authority in Menu Design

The availability of decision-support tools, such as Query/400 poses challenges for security design. You may want users to be able to view information in files using a query tool, but you probably want to make sure that the files are changed only by tested application programs.

No method exists in the resource security definitions for a user to have different authority to a file in different circumstances. However, using adopted authority allows you to define authority to meet different requirements.

Note: "Objects That Adopt the Owner's Authority" on page 5-6 describes how adopted authority works. Figure 5-11 on page 5-16 describes how the system checks for adopted authority.

Figure 7-5 shows a sample initial menu that uses adopted authority to provide controlled access to files using query tools:



```
MENU1          Initial Menu

          1. Inventory Control  (ICSTART)
          2. Customer Orders    (COSTART)
          3. Query              (QRYSTART)
          4. Office             (OFCSTART)

(no command line)
```

Figure 7-5. Sample Initial Menu

The programs that start applications (ICSTART and COSTART) adopt the authority of a profile that owns the application objects. The programs add application libraries tc the library list and display the initial application menu. Following is an example of the Inventory Control program (ICSTART).

```
PGM
ADDLIBLE ITEMLIB
ADDLIBLE ICPGMLIB
GO ICMENU
RMVLIBLE ITEMLIB
RMVLIBLE ICPGMLIB
ENDPGM
```

Figure 7-6. Sample Initial Application Program

The program that starts Query (QRYSTART) adopts the authority of a profile (QRYUSR) provided to allow access to files for queries. Figure 7-7 shows the QRYSTART program:

```
PGM
ADDLIBLE ITEMLIB
ADDLIBLE CUSTLIB
STRQRY
RMVLIBLE ITEMLIB
RMVLIBLE CUSTLIB
ENDPGM
```

Figure 7-7. Sample Program for Query with Adopted Authority

The menu system uses three types of user profiles, shown in Table 7-1. Table 7-2 describes the objects used by the menu system.

Table 7-1. User Profiles for Menu System

Profile Type	Description	Password	Limit Capabilities	Special Authorities	Initial Menu
Application owner	Owns all application objects and has *ALL authority. OWNIC owns Inventory Control application.	*NONE	N/A	As needed by application	N/A
Application user [1]	Example profile for anyone who uses the menu system	Yes	*YES	None	MENU1
Query Profile	Used to provide access to libraries for query	*NONE	N/A	None	N/A

[1] The current library specified in the application user profile is used to store any queries created. The Attention-key-handling program is *ASSIST, giving the user access to basic system functions.

Table 7-2. Objects Used by Menu System

Object Name	Owner	Public Authority	Private Authorities	Additional Information
MENU1 in QGPL library	See Note	*EXCLUDE	*USE authority for any users who are allowed to use the menu	In QGPL library because users do not have authority to application libraries
ICSTART program in QGPL	OWNIC	*EXCLUDE	*USE authority for users authorized to Inventory Control application	Created with USRPRF(*OWNER) to adopt OWNIC authority
QRYSTART program in QGPL	QRYUSR	*EXCLUDE	*USE authority for users authorized to create or run queries	Created with USRPRF(*OWNER) to adopt QRYUSR authority
ITEMLIB	OWNIC	*EXCLUDE	QRYUSR has *USE	
ICPGMLIB	OWNIC	*EXCLUDE		
Files available for Query in ITEMLIB	OWNIC	*USE		
Files not available for Query in ITEMLIB	OWNIC	*EXCLUDE		
Programs in ICPGMLIB	OWNIC	*USE		

Note: A special owner profile can be created for objects used by multiple applications.

When USERA selects option 1 (Inventory Control) from MENU1, program ICSTART runs. The program adopts the authority of OWNIC, giving *ALL authority to the inventory control objects in ITEMLIB and the programs in ICPGMLIB. USERA is thus authorized to make changes to the inventory control files while using options from the ICMENU.

When USERA exits ICMENU and returns to MENU1, the ITEMLIB and ICPGMLIB libraries are removed from the USERA library list, and program ICSTART is removed from the program stack. USERA is no longer running under adopted authority.

When USERA selects option 3 (Query) from MENU1, program QRYSTART runs. The program adopts the authority of QRYUSR, giving *USE authority to the ITEMLIB library. The public authority to the files in ITEMLIB determines which files USERA is allowed to query.

This technique has the advantage of minimizing the number of private authorities and providing good performance when checking authority:

- The objects in the application libraries do not have private authorities. For some application functions, public authority is adequate. If public authority is not adequate, owner authority is used. “Case 5: Public Authority without Private Authority” on page 5-22 shows the authority checking steps.

- Access to the files for query uses public authority to the files. The QRYUSR profile is only specifically authorized to the ITEMLIB library.
- By default, any query programs created are placed in the user's current library. The current library should be owned by the user, and the user should have *ALL authority.
- Individual users only need to be authorized to MENU1, ICSTART, and QRYSTART.

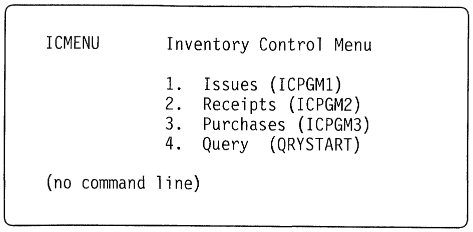
Consider these risks and precautions when using this technique:

- USERA has *ALL authority to all entire inventory control objects from ICMENU. Make sure the menu does not allow access to a command line or allow unwanted delete and update functions.
- Many decision-support tools allow access to a command line. The QRYUSR profile should be a limited capability user without special authorities to prevent unauthorized functions.

Ignoring Adopted Authority: “Using Adopted Authority in Menu Design” shows a technique for providing query capability without allowing uncontrolled changes to application files. This technique requires the user to return to the initial menu before running queries. If you want to provide the convenience of starting query from application menus as well as from the initial menu, you can set up the QRYSTART program to ignore adopted authority.

Note: “Programs That Ignore Adopted Authority” on page 5-8 provides more information about ignoring adopted authority. Figure 5-11 on page 5-16 describes how the system checks for adopted authority.

Figure 7-8 shows an application menu that includes the QRYSTART program:



```
ICMENU       Inventory Control Menu

             1.  Issues (ICPGM1)
             2.  Receipts (ICPGM2)
             3.  Purchases (ICPGM3)
             4.  Query  (QRYSTART)

(no command line)
```

Figure 7-8. Sample Application Menu with Query

The authority information for the QRYSTART program is the same as shown in Table 7-2 on page 7-5. The program is created with the use adopted authority (USEADPAUT) parameter set to *NO, to ignore the adopted authority of previous programs in the stack.

Following are comparisons of the program stacks when USERA selects query from MENU1 (see Figure 7-5 on page 7-4) and from ICMENU:

Program stack when query selected from MENU1

MENU1 (no adopted authority)
QRYSTART (adopted authority QRYUSR)

Program stack when query selected from ICMENU

MENU1 (no adopted authority)
ICMENU (adopted authority OWNIC)
QRYSTART (adopted authority QRYUSR)

By specifying the QRYSTART program with USEADPAUT(*NO), the authority of any previous programs in the stack is not used. This allows USERA to run query from ICMENU without having the ability to change and delete files, because the authority of OWNIC is not used by the QRYSTART program.

When USERA ends query and returns to ICMENU, adopted authority is once again active. Adopted authority is ignored only as long as the QRYSTART program is active.

If public authority to the QRYSTART program is *USE, specify USEADPAUT(*NO) as a security precaution. This prevents anyone running under adopted authority from calling the QRYSTART program and performing unauthorized functions.

The inquiry menu (Figure 7-4 on page 7-4) at the JKL Toy Company also uses this technique, because it can be called from menus in different application libraries. It adopts the authority of QRYUSR and ignores any other adopted authority in the program stack.

Describing Menu Security

As an application designer, you need to provide information about a menu for the security administrator. The security administrator uses this information to decide who should have access to the menu and what authorities are required. Typical information needed is:

- Whether any menu options require special authorities, such as *SAVSYS or *JOBCTL.
- Whether menu options call programs that adopt authority.
- What authority to objects is required for each menu option. You should only need to identify those authorities that are greater than normal public authority.

Figure 7-9 on page 7-7 shows a sample format for providing this information.

Menu name: MENU1 Library: QGPL

Option number: 3 Description: Query

Program called: QRYSTART Library: QGPL

Authority adopted: QRYUSR

Special authority required: None

Object authorities required: User must have *USE authority to QRYSTART program. QRYUSR must have *USE authority to libraries containing files to be queried. User, QRYUSR, or public must have *USE authority to files being queried.

Figure 7-9. Format for Menu Security Requirements

System Request Menu

A user can use the system request function to suspend the current job and display the System Request Menu. The System Request Menu allows the user to send and display messages, transfer to a second job, or end the current job. Options on the System Request Menu are described in detail in the *New User's Guide.*

When your system is shipped, public authority to the System Request Menu is *USE. The simplest way to prevent users from accessing this menu is by restricting authority to the panel group QGMNSYSR:

- To prevent specific users from seeing the System Request Menu, specify *EXCLUDE authority for those users:

```
GRTOBJAUT OBJ(QSYS/QGMNSYSR) +
          OBJTYPE(*PNLGRP)   +
          USER(USERA) AUT(*EXCLUDE)
```

- To prevent most users from seeing the System Request Menu, revoke public authority and grant *USE authority to specific users:

```
RVKOBJAUT OBJ(QSYS/QGMNSYSR) +
          OBJTYPE(*PNLGRP) +
          USER(*PUBLIC) AUT(*ALL)
GRTOBJAUT OBJ(QSYS/QGMNSYSR) +
          OBJTYPE(*PNLGRP)   +
          USER(USERA) AUT(*USE)
```

You can prevent users from selecting specific options from the System Request Menu by restricting the authority to the associated commands. Table 7-3 shows the commands associated with the menu options:

Table 7-3. Options and Commands for the System Request Menu

Option	Command
1	Transfer Secondary Job (TFRSECJOB)
2	End Request (ENDRQS)
3	Display Job (DSPJOB)
4	Display Message (DSPMSG)
5	Send Message (SNDMSG)
6	Display Message (DSPMSG)
7	Display Work Station User (DSPWSUSR)
10	See note below
11	See note below
12	Display 3270 emulation options (See note below.)
80	Disconnect Job (DSCJOB)
90	Sign-Off (SIGNOFF)

Notes:

1. Options 10 and 11 are only displayed if display station pass-through has been started with the Start Pass-Through (STRPASTHR) command. Option 10 is only displayed on the target system.
2. Option 12 is only displayed when 3270 emulation is active.
3. Some of the options have restrictions for the System/36 environment. See the *Concepts and Programmer's Guide for the System/36 Environment* for more information about these restrictions.

For example, to prevent users from transferring to an alternative interactive job, revoke public authority to the Transfer to Secondary Job (TFRSECJOB) command and grant authority only to specific users:

```
RVKOBJAUT OBJ(TFRSECJOB) OBJTYPE(*CMD)
          USER(*PUBLIC) AUT(*ALL)
GRTOBJAUT OBJ(TFRSECJOB) OBJTYPE(*CMD)
          USER(USERA) AUT(*USE)
```

If a user selects an option for which the user does not have authority, a message is displayed.

If you want to prevent users from general use of the commands from the System Request menu but still want them to be able to run a command at a specific time (such as sign-off), you can create a CL program that adopts the authority of an authorized user and runs the command.

Planning Command Security

Menu security is a good technique for users who need applications and limited system functions. Some users need a more flexible environment and the capability to run commands. When your system arrives, the ability to use commands is set up to meet the security needs of most installations. Some commands can be run only by a security officer. Others require a special authority, such as *SAVSYS. Most commands can be used by anyone on the system.

You can change the authority to commands to meet your security requirements. For example, you may want to prevent most users on your system from working with communications. You can set the public authority to *EXCLUDE for all commands that work with communications objects, such the CHGCTLxxx, CHGLINxxx, and CHGDEVxxx commands.

If you need to control which commands can be run by users, you can use object authority to the commands themselves. Every command on the system has object type *CMD and can be authorized to the public or only to specific users. To run a command, the user needs *USE authority to it. Appendix C lists all the commands that are shipped with the public authority set to *EXCLUDE.

If you use the System/38 library, you need to restrict security-relevant commands in that library also. Or, you could restrict access to the entire library. If you use one or more national language versions of the OS/400 licensed program on your system, you need to restrict commands in the additional QSYSxxx libraries on your system as well.

Another useful security measure is to change the default values for some commands. The Change Command Default (CHGCMDDFT) command allows you to do this.

Planning File Security

The information contained in database files is usually the most important asset on your system. Resource security allows you to control who can view, change, and delete information in a file. If users require different authority to files depending on the situation, you can use adopted authority. "Using Adopted Authority in Menu Design" on page 7-4 gives an example of this method.

For critical files on your system, keep a record of what users have authority to the file. If you use group authority and authorization lists, you need to keep track of users who have authority through those methods, as well as users who are directly authorized. If you use adopted authority, you can list programs that adopt the authority of a particular user using the Display Program Adopt (DSPPGMADP) command.

You can also use the journaling function on the system to monitor activity against a critical file. Although the primary intent of a journal is to recover information, it can be used as a security tool. It contains a record of who has accessed a file and in what way. You can use the Display Journal (DSPJRN) command to view a sampling of journal entries periodically.

Logical Files

Resource security on the system does not directly support field-level or record-level security of a file. However, you can design logical files to protect particular fields or records in a file.

A logical file can be used to specify a subset of *records* that a user can access (by using select and omit logic). Therefore, specific users can be prevented from accessing certain record types. A logical file can be used to specify a subset of *fields* in a record that a user can access. Therefore, specific users can be prevented from accessing certain fields in a record.

A logical file does not contain any data. It is a particular view of one or more physical files that contain the data. Because a logical file does not contain data, you can specify only object management authorities, not data authorities, for a logical file. Providing access to the information defined by a logical file requires a combination of object management authority to the logical file and data authority to the associated physical files.

Figure 7-10 on page 7-9 illustrates how to use a logical file to restrict access to certain fields. For example, to give the user the ability to change fields FLDA and FLDB in FILEAP, but not to add or delete records in the file, the user needs *OBJOPR authority to the logical file (FILEAL). The user needs *READ and *UPD authority to the physical file (FILEAP). The user should *not* have *OBJOPR authority to the physical file. This would allow the user to view and update the physical file directly, including field FLDC.

(1) The user must have *OBJOPR authority to open the logical file. *OBJOPR authority to a file allows the user to perform all functions allowed by the user's data authorities to the physical file.

(2) Because the logical file format does not include FLDC, the user cannot access that field through the logical file.

(3) For the physical file, the user needs any data authorities necessary to perform the operations in PGM1. For example, if the program allows deletion of records in the logical file (FILEAL), the user must have *DLT authority to the physical file (FILEAP).

(4) To prevent the user from directly accessing FILEAP and thus having access to FLDC, the user should not have *OBJOPR authority to FILEAP. The user cannot open physical file FILEAP without *OBJOPR.

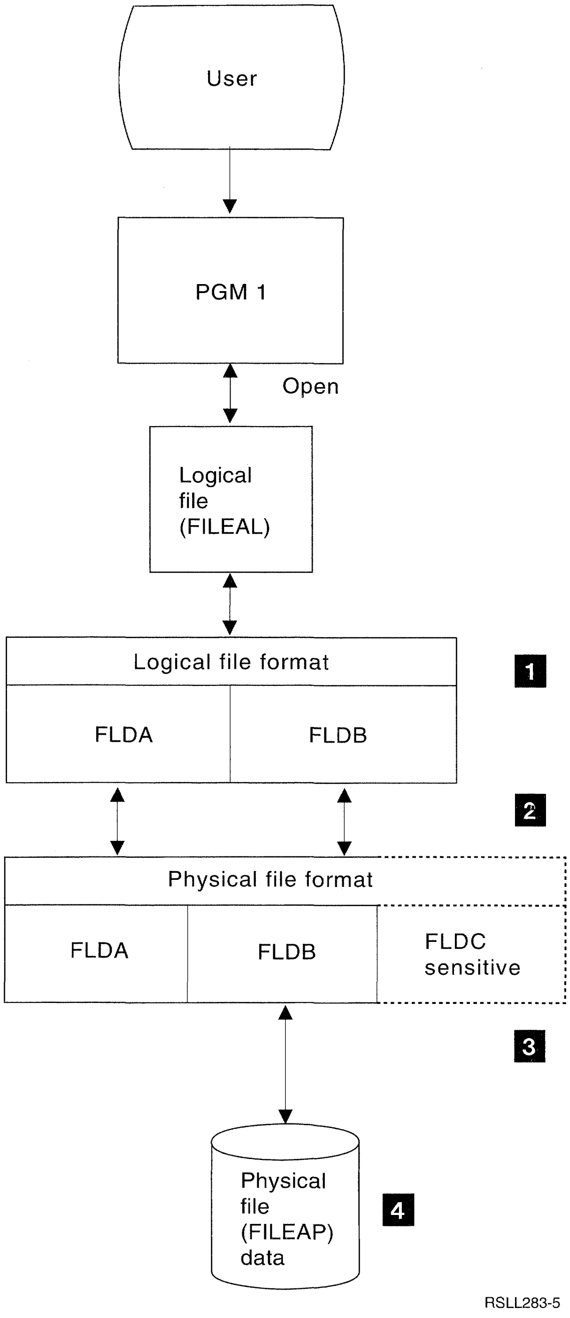


Figure 7-10. Using a Logical File for Security

Overriding Files

Override commands can be used to have a program use a different file with the same format. For example, assume that a program in the contracts and pricing application at the JKL Toy Company writes pricing information to a work file before making price changes. A user with access to a command line who wanted to capture confidential information could use an override command to cause the program to write data to a different file in a library controlled by the user.

You can make sure a program processes the correct files by using override commands with SECURE(*YES) before the program runs.

Planning Authorization Lists

An authorization list has these advantages:

- Authorization lists simplify managing authorities. User authority is defined for the authorization list, not for the individual objects on the list. If a new object is secured by the authorization list, the users on the list gain authority to the object.
- One operation can be used to give a user authority to all the objects on the list.
- Authorization lists reduce the number of private authorities on the system. Each user on the list has a private authority to one object, the authorization list. This gives the user authority to all the objects on the list.
- Authorization lists provide a way to remember authorities when an object is saved. When an object is saved that is secured by an authorization list, the name of the authorization list is saved with the object. If the object is deleted and restored to the **same** system, it is automatically linked to the authorization list again. If the object is restored on a different system, the authorization list is not linked, unless ALWOBJDIF(*ALL) is specified on the restore command.

Use authorization lists to secure objects with similar requirements. Consider the advantages of the authorization list and whether they are a genuine benefit in managing the objects in question. If only a few objects would be secured by the list, the additional overhead in performance checking may not balance the reduction in the number of private authorities on the system.

At the JKL Toy Company, an authorization list is used to secure all the work files used in month-end inventory processing. These work files are cleared, which requires *OBJMGT authority. As application requirements change, more work files may be added to the application. Also, as job responsibilities change, different users run month-end processing. An authorization list makes it simpler to manage these changes.

Following are the steps to set up the authorization list:

1. Create the authorization list:

```
CRTAUTL ICLIST1
```

2. Secure all the work files with the authorization list:

```
GRTOBJAUT OBJ(ITEMLIB/ICWRK*) +
          OBJTYP(*FILE) AUTL(ICLIST1)
```

3. Add users to the list who perform month-end processing:

```
ADDAUTLE AUTL(ICLIST1) USER(USERA) AUT(*ALL)
```

Authorization Lists and Referenced Objects

Sometimes defining private authorities for objects with similar security requirements is preferable to using an authorization list. The referenced object (REFOBJ) parameter on the GRTOBJAUT command and the function key on the Edit Object Authority display provide a method for copying private authorities from one object to another.

From a security management standpoint, an authorization list is the preferred method for handling objects with the same security requirements. An authorization list is easier to manage and results in fewer private authorities on the system. However, securing an object with an authorization list causes two searches of the user's private authorities during authority checking and may cause performance problems.

For example, assume you have ten files and need to give identical authority to three users:

- If you grant authority to each file, using the referenced object technique, this results in 30 private authorities (ten for each of three users) on the system.
- If you secure the ten files with an authorization list, you create one additional object on the system (the authorization list), but you only need three private authorities. Each of three users has authority to the authorization list.
- When the system checks authority, the user's profile is searched first for authority to the object and then for authority to the authorization list. See Figure 5-7 on page 5-12 for details.

The number of private authorities on the system affects the time it takes to save the system (SAVSYS) or to save security data (SAVSECDTA). However, using an authorization list results in searching the user's private authority twice. These two things need to be balanced.

Planning Group Profiles

A group profile is a useful tool when several users have similar security requirements. They are particularly useful when job requirements and group membership change. For example, if members of a department have responsibility for an application, a group profile can be set up for the department. As users join or leave the department, the group profile field in their user profiles can be changed. This is easier to manage than removing individual authorities from user profiles.

You can create profiles specifically to be group profiles, or you can make an existing profile into a group profile. A group profile is simply a special type of user profile. It becomes a group profile when another profile designates it as the group profile. For example:

1. Create a profile called GRPIC:

   ```
   CRTUSRPRF GRPIC
   ```
2. When the profile is created, it is an ordinary profile, not a group profile.
3. Designate GRPIC as the group profile for another group profile:

   ```
   CHGUSRPRF USERA GRPPRF(GRPIC)
   ```
4. The system now treats GRPIC as a group profile.

Creating profiles specifically to be group profiles is preferable to making existing profiles into group profiles. You may find that a specific user has all the authorities needed by a group of users and be tempted to make that user profile into a group profile. However, using an individual's profile as a group profile may cause problems in the future:

- If the user whose profile is used as the group profile changes responsibilities, a new profile needs to be designated as the group profile, authorities need to be changed, and object ownership needs to be transferred.
- All members of the group automatically have authority to any objects created by the group profile. The user whose profile is the group profile loses the ability to have private objects, unless that user specifically excludes other users.

Try to plan group profiles in advance. Create specific group profiles with password *NONE. If you discover after an application has been running that a user has authorities that should belong to a group of users, do the following:

1. Create a group profile.
2. Use the GRTUSRAUT command to give the user's authorities to the group profile.
3. Remove the private authorities from the user, because they are no longer needed. Use the RVKOBJAUT or EDTOBJAUT command.

Comparison of Group Profiles and Authorization Lists

Group profiles are used to simplify managing user profiles for those who have similar security requirements. Authorization lists are used to secure objects with similar security requirements. Table 7-4 shows the characteristics of the two methods:

Table 7-4 (Page 1 of 2). Authorization List and Group Profile Comparison

Item Being Compared	Authorization List	Group Profile
Used to secure multiple objects	Yes	Yes
User can belong to more than one	Yes	No
Private authority overrides other authority	Yes	Yes

Table 7-4 (Page 2 of 2). Authorization List and Group Profile Comparison

Item Being Compared	Authorization List	Group Profile
User must be assigned authority independently	Yes	No
Authorities specified are the same for all objects	Yes	No
Object can be secured by more than one	No	Yes
Authority can be specified when the object is created	Yes	Yes [1]
Can secure all object types	No	Yes
Association with object is deleted when the object is deleted	Yes	Yes
Association with object is saved when the object is saved	Yes	No

[1] Members of the group profile can be given authority at the time an object is created by the GRPAUT parameter in the profile of the user creating an object.

Planning Security for Programmers

Programmers pose a problem for the security officer. Their knowledge makes it possible for them to bypass security procedures that are not carefully designed. They can bypass security to access data they need for testing. They can also circumvent the normal procedures that allocate system resources in order to achieve better performance for their own jobs. Security is often seen by them as a hindrance to doing the tasks required by their job, such as testing applications. However, giving programmers too much authority on the system breaks the security principle of separating duties. It also allows a programmer to install unauthorized programs.

Follow these guidelines when setting up an environment for application programmers:

- Do not grant *ALLOBJ, *SERVICE, *SAVSYS, *AUDIT, or *SECADM special authority to the programmer.
- Do not use the QPGMR user profile as a group profile for programmers.
- Use test libraries and prevent access to production libraries.
- Create programmer libraries and use a program that adopts authority to copy selected production data to programmer libraries for testing.
- If interactive performance is an issue, consider changing the commands for creating programs to run only in batch:

  ```
  CHGCMD CMD(CRTxxxPGM) ALLOW(*BATCH *BPGM)
  ```

- Perform security auditing of application function before moving applications or program changes from test to production libraries.
- Use the group profile technique when an application is being developed. Have all application programs owned by a group profile. Make programmers who work on the application members of the group and define the programmer user profiles to have the group own any new objects created (OWNER(*GRPPRF)). When a programmer moves from one project to another, you can change the group information in the programmer's profile. See "Group Ownership of Objects" on page 5-6 for more information.
- Develop a plan for assigning ownership of applications when they are moved into production. To control changes to a production application, all application objects, including programs, should be owned by the user profile designated for the application.

  Application objects should not be owned by a programmer because the programmer would have uncontrolled access to them in a production environment. The profile that owns the application may be the profile of the individual responsible for the application, or it may be a profile specifically created as the application owner. The Change Library Owner (CHGLIBOWN) tool in the QUSRTOOL library can assist you with object ownership.

Managing Source Files

Source files are important to the integrity of your system. They may also be a valuable company asset, if you have developed or acquired custom applications. Source files should be protected like any other important file on the system. Consider placing source files in separate libraries and controlling who can update them and move them to production.

When a source file is created on the system, the default public authority is *CHANGE, which allows any user to update any source member. By default, only the owner of the source file or a user with *ALLOBJ special authority can add or remove members. In most cases, this default authority for source physical files should be changed. Programmers working on an application need *OBJMGT authority to the source files to add new members. The public authority should probably be reduced to *USE or *EXCLUDE, unless the source files are in a controlled library.

Planning Security for System Programmers or Managers

Most systems have someone responsible for housekeeping functions. This person monitors the use of system resources, particularly disk storage, to make sure that users regularly remove unused objects to free space. System programmers need broad authority to observe all the objects on the system. However, they do not need to view the contents of those objects.

You can use adopted authority to provide a set of display commands for system programmers, rather than giving special authorities in their user profiles. The Display with Adopt (DSPADP) tool in the QUSRTOOL library provides a set of commands for use by the system programmer.

Chapter 8. Backup and Recovery

This chapter discusses how security relates to backup and recovery on your system:

- How security information is saved and restored
- How security affects saving and restoring objects
- Security issues associated with *SAVSYS special authority

The *Basic Backup and Recovery Guide* and the *Advanced Backup and Recovery Guide* provide more information about backup and recovery. The *Office Services Concepts and Programmer's Guide* provides information about saving and restoring OfficeVision/400 objects.

Saving your security information is just as important as saving your data. In some situations, you may need to recover user profiles, object authorities, and the data on your system. If you do not have your security information saved, you may need to manually rebuild user profiles and object authorities. This can be time-consuming and can lead to errors and security exposures.

Planning adequate backup and recovery procedures for security information requires understanding how the information is stored, saved, and restored.

Table 8-1 shows the commands used to save and restore security information. The sections that follow discuss saving and restoring security information in more detail.

Table 8-1. How Security Information Is Saved and Restored

	Save and Restore Commands Used				
Security Information Saved or Restored	**SAVSECDTA** **SAVSYS**	**SAVCHGOBJ** **SAVOBJ** **SAVLIB** **SAVDLO** **SAVCFG**	**RSTUSRPRF**	**RSTOBJ** **RSTLIB** **RSTDLO** **RSTCFG**	**RSTAUT**
User profiles	X		X		
Object ownership [1]		X		X	
Public authorities [1]		X		X	
Private authorities	X				X
Authorization lists	X		X		
Authority holders	X		X		
Link with the authorization list and authority holders		X		X	
Object auditing value		X		X	

[1] The SAVSECDTA, SAVSYS, and RSTUSRPRF commands save and restore ownership and public authority for these object types:

User profile (*USRPRF)
Authorization list (*AUTL)
Authority holder (*AUTHLR)

How Security Information Is Stored

Security information is stored with objects, user profiles, and authorization lists:

Authority Information Stored with Object

Public authority
Owner name
Owner's authority to object
Authorization list name
Object auditing value
Whether any private authority exists
Whether any private authority is less than public

Authority Information Stored with User Profile

Heading Information:
The user profile attributes shown on the Create User Profile display.

Private Authority Information:
Private authority to objects. This includes private authority to authorization lists.

Ownership Information:
List of owned objects
For each owned object, a list of users with private authority to the object.

Auditing Information:
Action auditing value
Object auditing value

Authority Information Stored with Authorization Lists

Normal authority information stored with any object, such as the public authority and owner.

List of all objects secured by the authorization list.

Saving Security Information

Security information is stored differently on the save media than it is on your system. When you save user profiles, the private authority information stored with the user profile is formatted into an authority table. An authority table is built and saved for each user profile that has private authorities. This reformatting and saving of security information can be lengthy if you have many private authorities on your system.

This is how security information is stored on the save media:

Authority Information Saved with Object

Public authority
Owner name
Owner's authority to object
Authorization list name
Object auditing value
Whether any private authority exists
Whether any private authority is less than public

Authority Information Saved with Authorization List

Normal authority information stored with any object, such as the public authority and owner.

Authority Information Saved with User Profile

Heading Information:
The user profile attributes shown on the Create User Profile display.

Authority Table Saved Associated with User Profile

One record for each private authority the user profile has.

Recovering Security Information

Recovering your system often requires restoring data and associated security information. The usual sequence for recovery is:

1. Restore user profiles and authorization lists (RSTUSRPRF USRPRF(*ALL)).
2. Restore objects (RSTLIB, RSTOBJ, or RSTCFG).
3. Restore the private authorities to objects (RSTAUT).

The *Basic Backup and Recovery Guide* provides more information about planning recovery.

Restoring User Profiles

Some changes may be made to a user profile when it is restored. The following applies:

- If profiles are being restored individually (RSTUSRPRF USRPRF(*ALL) is not specified) and the profile being restored does not exist on the system, these fields are changed to *NONE:
  - Group profile name (GRPPRF)
  - Password (PASSWORD)
  - Document password (DOCPWD)
- If profiles are being restored individually (RSTUSRPRF USRPRF(*ALL) is not specified) and the profile exists or the system, the password, document password, and group profile are not changed.
- If all user profiles are being restored to your system, all the fields in any profiles that already exist on the system are restored from the save media, including the password.

  Warning: Keep a record of the security officer (QSECOFR) password associated with each version of your security information that is saved to make sure you can sign on to your system if you need to do a complete restore operation.

  You can use DST (Dedicated Service Tools) to reset the password for the QSECOFR profile. See "Resetting the QSECOFR Password" on page 4-24 for instructions.
- *ALLOBJ special authority is removed from user profiles being restored to a system at security level 30 or higher in either of these situations:
  - The profile was saved from a different system.
  - The profile was saved from the same system at security level 10 or 20.

  Note: The system uses the machine serial number on the system and on the save media to determine whether objects are being restored to the same system or a different system.

  *ALLOBJ special authority is not removed from these IBM-supplied profiles:

  QSYS (system) user profile
  QSECOFR (security officer) user profile
  QLPAUTO (licensed program automatic install) user profile
  QLPINSTALL (licensed program install) user profile

Restoring Objects

When you restore an object to the system, the system uses the authority information stored with the object. The following applies to security of the restored object:

Object ownership

- If the profile that owns the object is on the system, ownership is restored to that profile.

- If the owner profile does not exist on the system, ownership of the object is given to the QDFTOWN (default owner) user profile.
- If the object exists on the system and the owner on the system is different from the owner on the save media, the object is not restored unless ALWOBJDIF(*ALL) is specified. In that case, the object is restored and the owner on the system is used.
- See "Restoring Programs" on page 8-4 for additional considerations when restoring programs.

Public authority

- If the object being restored does not exist on the system, public authority is set to the public authority of the saved object.
- If the object being restored does exist and is being replaced, public authority is not changed. The public authority from the saved version of the object is not used.
- The CRTAUT for the library is not used when restoring objects to the library.

Authorization list

- If an object, other than a document or folder, already exists on the system and is linked to an authorization list, it must have the same authorization list as the saved object. If not, the object is not restored.
- If a document or folder that already exists on the system is restored, the authorization list associated with the object on the system is used. The authorization list from the saved document or folder is not used.
- If the authorization list does not exist on the system, the object is restored without being linked to an authorization list and the public authority is changed to *EXCLUDE.
- If the object is being restored on the same system from which it was saved, the object is linked to the authorization list again.
- If the object is being restored on a different system, the ALWOBJDIF parameter on the restore command is used to determine whether the object is linked to the authorization list:
  - If ALWOBJDIF(*ALL) is specified, the object is linked to the authorization list.
  - If ALWOBJDIF(*NONE) is specified, then the object is not linked to the authorization list and the public authority of the object is changed to *EXCLUDE.

Private authorities

- Private authority is saved with user profiles, not with objects.
- If user profiles have private authority to an object being restored, those private authorities are usually not affected. Restoring certain types of programs may result in private authorities being revoked. See "Restoring Programs" on page 8-4 for more information.
- If an object is deleted from the system and then restored from a saved version, private authority for the object no longer exists on the system. When an object is deleted, all private authority to the object is removed from user profiles.
- If private authorities need to be recovered, the Restore Authority (RSTAUT) command must be used. The normal sequence is:
  1. Restore user profiles
  2. Restore objects
  3. Restore authority

Object Auditing

- If the object being restored does not exist on the system, the object auditing (OBJAUD) value of the saved object is restored.
- If the object being restored does exist and is being replaced, the object auditing value is not changed. The OBJAUD value of the saved version of the object is not restored.
- If a library being restored does not exist on the system, the create object auditing (CRTOBJAUD) value for the library is restored.
- If a library being restored exists and is being replaced, the CRTOBJAUD value for the library is not restored. The CRTOBJAUD value for the existing library is used.

Authority Holder

- If a file is restored and an authority holder exists for that file name and the library to which it is being restored, the file is linked to the authority holder.
- The authority information associated with the authority holder replaces the public authority and owner information saved with the file.

User Domain Objects

- For systems running Version 2 Release 3 or later of the OS/400 licensed program, the system restricts user domain objects (*USRSPC, *USRIDX, and *USRQ) to the libraries specified in the QALWUSRDMN system value. If a library is removed from the QALWUSRDMN system value after a user domain object of type *USRSPC, *USRIDX, or *USRQ is saved, the system changes the object to system domain when it is restored.

  Note: A system domain object of type *USRSPC, *USRIDX, or *USRQ cannot be restored to a system running a version of the OS/400 licensed program earlier than Version 2 Release 3.

Restoring Authority

When security information is restored, private authorities must be rebuilt. When you restore a user profile that has an authority table, the authority table for the profile is also restored.

The Restore Authority (RSTAUT) command rebuilds the private authority in the user profile using the information from the authority table. The grant authority operation is run for each private authority in the authority table. If authority is being restored for many profiles and many private authorities exist in the authority tables, this can be a lengthy process.

Although profiles cannot be saved individually, they can be restored individually. The RSTUSRPRF and RSTAUT commands can be run for a single profile, a list of profiles, a generic profile name, or all profiles. The system searches the save media or save file created by the SAVSECDTA or SAVSYS command to find the profiles you want to restore.

Restoring Programs

Restoring programs to your system from a different source poses a security exposure. Programs might perform operations that break your security requirements. Of particular concern are programs that contain restricted instructions and programs that adopt owner authority. The system performs special checking when these types of programs are restored to your system.

To protect against programs that contain restricted instructions, the system uses a validation value. This value is stored with a program and recalculated when the program is restored. The system's actions are determined by the ALWOBJDIF parameter on the restore command and by the security level (QSECURITY) system value. "Validation of Programs Being Restored" on page 2-6 provides a detailed chart and description of the alternatives.

Restoring Programs That Adopt the Owner's Authority: When a program is restored that adopts owner authority, the ownership and authority to the program may be changed. The following applies:

- The user profile doing the restore operation must either own the program or have *ALLOBJ and *SECADM special authorities.
- The user profile doing the restore operation can receive the authority to restore the program by
  - Being the program owner.
  - Being a member of the group profile that owns the program (unless you have private authority to the program).
  - Having *ALLOBJ and *SECADM special authority.
  - Being a member of a group profile that has *ALLOBJ and *SECADM special authority.
  - Running under adopted authority that meets one of the tests just listed.
- If the restoring profile does not have adequate authority, all public and private authorities to the program are revoked, and the public authority is changed to *EXCLUDE.
- If the owner of the program does not exist on the system, ownership is given to the QDFTOWN user profile. Public authority is changed to *EXCLUDE and the authorization list is removed.

Restoring Licensed Programs

The Restore Licensed Programs (RSTLICPGM) command is used to install IBM-supplied programs on your system. It can also be used to install non-IBM programs created using the SAA** SystemView* System Manager/400* licensed program

When your system is shipped, only users with *ALLOBJ special authority can use the RSTLICPGM command. The RSTLICPGM procedure calls an exit program to install programs that are not supplied by IBM.

To protect security on your system, the exit program should not run using a profile with *ALLOBJ special authority. Use a program that adopts *ALLOBJ special authority to run the RSTLICPGM command, instead of having a user with *ALLOBJ authority run the command directly.

Following is an example of this technique. The program to be installed using the RSTLICPGM command is called CPAPP (Contracts and Pricing).

1. Create a user profile with sufficient authority to successfully install the application. Do not give this profile *ALLOBJ special authority. For the example, the user profile is called OWNCP.
2. Write a program to install the application. For the example, the program is called CPINST:

   ```
   PGM
   RSTLICPGM CPAPP
   ENDPGM
   ```

3. Create the CPINST program to adopt the authority of a user with *ALLOBJ special authority, such as QSECOFR, and authorize OWNCP to the program:

   ```
   CRTCLPGM QGPL/CPINST USRPRF(*OWNER) +
            AUT(*EXCLUDE)
   GRTOBJAUT OBJ(CPINST) OBJTYP(*PGM) +
             USER(OWNCP)  AUT(*USE)
   ```

4. Sign on as OWNCP and call the CPINST program. When the CPINST program runs the RSTLICPGM command, you are running under QSECOFR authority. When the exit program runs to install the CPAPP programs, it drops adopted authority. The programs called by the exit program run under the authority of OWNCP.

Restoring Authorization Lists

Authorization lists are saved by either the SAVSECDTA command or the SAVSYS command. Authorization lists are restored by the command:

```
RSTUSRPRF USRPRF(*ALL)
```

No method exists for restoring an individual authorization list.

When you restore an authorization list, authority and ownership are established just as they are for any other object that is restored. The link between authorization lists and objects is established if the objects are restored after the authorization list. See "Restoring Objects" on page 8-2 for more information. Users' private authorities to the list are restored using the RSTAUT command.

Recovering from a Damaged Authorization List:

When an object is secured by an authorization list and the authorization list becomes damaged, access to the object is limited to users that have all object (*ALLOBJ) special authority.

To recover from a damaged authorization list, two steps are required:

1. Recover users and their authorities on the authorization list.
2. Recover the association of the authorization list with the objects.

These steps must be done by a user with *ALLOBJ special authority.

Recovering the Authorization List: If users' authorities to the authorization list are known, simply delete the authorization list, create the authorization list again, and then add users to it.

If it is not possible to create the authorization list again because you do not know all the user authorities, the authorization list can be restored and the users restored to the authorization list using your last SAVSYS or SAVSECDTA tapes. To restore the authorization list, do the following:

1. Delete the damaged authorization list using the Delete Authorization List (DLTAUTL) command.
2. Restore the authorization list by restoring user profiles:

   ```
   RSTUSRPRF USRPRF(*ALL)
   ```

3. Restore users' private authorities to the list using the RSTAUT command.

Warning: This procedure restores user profile values from the save media. See "Restoring User Profiles" on page 8-2 for more information.

Recovering the Association of Objects to the Authorization List: When the damaged authorization list is deleted, the objects secured by the authorization list need to be added to the new authorization list. Do the following:

1. Find the objects that were associated with the damaged authorization list using the Reclaim Storage (RCLSTG) command. Reclaim storage assigns the objects that were associated with the authorization list to the QRCLAUTL authorization list.
2. Use the Display Authorization List Objects (DSPAUTLOBJ) command to list the objects associated with the QRCLAUTL authorization list.
3. Use the Grant Object Authority (GRTOBJAUT) command to secure each object with the correct authorization list:

   ```
   GRTOBJAUT OBJ(library-name/object-name) +
             OBJTYPE(object-type) +
             AUTL(authorization-list-name)
   ```

Note: If a large number of objects are associated with the QRCLAUTL authorization list, create a database file by specifying OUTPUT(*OUTFILE) on the DSPAUTLOBJ command. You can write a CL program to run the GRTOBJAUT command for each object in the file.

Restoring the Operating System

When you perform a manual IPL on your system, the IPL or Install the System menu provides an option to install the operating system. The dedicated service tools (DST) function provides the ability to require anyone using this menu option to enter the DST security password. You can use this to prevent someone from restoring an unauthorized copy of the operating system.

To secure the installation of your operating system, do the following:

1. Perform a manual IPL.
2. From the IPL or Install the System menu, select DST.
3. From the Use DST menu, select the option to work with the DST environment.
4. Select the option to change DST passwords.
5. Select the option to change the operating system install security.
6. Specify 1 (secure).
7. Press F3 (exit) until you return to the IPL or Install the System menu.
8. Complete the manual IPL and return the keylock to its normal position.

Notes:

1. If you no longer want to secure the installation of the operating system, follow the same steps and specify 2 (not secure).
2. You can also prevent installation of the operating system by keeping your keylock switch in the normal position and removing the key.

*SAVSYS Special Authority

To save or restore an object, you must have *OBJEXIST authority to the object or *SAVSYS special authority. A user with *SAVSYS special authority does not need any additional authority to an object to save or restore it.

*SAVSYS special authority gives a user the capability to save an object and take it to a different system to be restored or to display (dump) the media to view the data. It also gives a user the capability to save an object and free storage thus deleting the data in the object. When saving documents, a user with *SAVSYS special authority has the option to delete those documents. *SAVSYS special authority should be given carefully.

The Restore Any Library (RSTANYLIB) and Restore Any File (RSTANYFIL) commands in the QUSRTOOL library show examples of using adopted authority instead of giving *SAVSYS special authority to system operators.

Auditing Save and Restore Operations

A security audit record is written for each restore operation if the action auditing value (QAUDLVL system value or AUDLVL in the user profile) includes *SAVRST. When you use a command that restores a large number of objects, such as RSTLIB, an audit record is written for each object restored. This may cause problems with the size of the audit journal receiver, particularly if you are restoring more than one library.

The RSTCFG command does not create an audit record for each object restored. If you want to have an audit record of this command, set object auditing for the command itself. One audit record will be written whenever the command is run.

The *SAVRST audit level does not apply to save operations. You can monitor save operations in two ways:

- Specify object auditing for the SAVxxx commands.
- Specify object auditing for specific objects, if you want the security audit journal to show that they have been saved.

Commands that save a very large number of objects, such as SAVSYS, SAVSECDTA, and SAVCFG, do not create individual audit records for the objects saved, even if the saved objects have object auditing active. To monitor these commands, set up object auditing for the commands themselves

Chapter 9. Auditing Security on the AS/400 System

This chapter describes techniques for auditing the effectiveness of security on your system. People audit their system security for several reasons:

- To evaluate whether the security plan is complete.
- To make sure that the planned security controls are in place and working. This type of auditing is usually performed by the security officer as part of daily security administration. It is also performed, sometimes in greater detail, as part of a periodic security review by internal or external auditors.
- To make sure that system security is keeping pace with changes to the system environment. Some examples of changes that affect security are:
  - New objects created by system users
  - New users admitted to the system
  - Change of object ownership (authorization not adjusted)
  - Change of responsibilities (user group changed)
  - Temporary authority (not timely revoked)
  - New products installed
- To prepare for a future event, such as installing a new application, moving to a higher security level, or setting up a communications network.

The techniques described in this chapter are appropriate for all these situations. Which things you audit and how often depends on the size and security needs of your organization. The purpose of this chapter is to discuss what information is available, how to obtain it, and why it is needed, rather than to give guidelines for the frequency of audits.

This chapter has three parts:

- A checklist of security items that can be planned and audited.
- Information about setting up and using the audit journal provided by the system.
- Other techniques that are available to gather security information on the system.

Security auditing involves using commands on the AS/400 system and accessing log and journal information on the system. Some of the auditing tasks suggested in this chapter require a user profile with *ALLOBJ and *SECADM special authority. Others require *AUDIT special authority.

Checklist for Security Officers and Auditors

This checklist can be used both to plan and to audit system security. As you plan security, choose the items from the list that meet your security requirements. When you audit the security of your system, use the list to evaluate the controls you have in place and to determine if additional controls are needed.

This list serves as a review of the information in this manual. The list contains brief descriptions of how to do each item and how to monitor that it has been done, including what entries in the QAUDJRN journal to look for. Details about the items are found throughout the manual.

Physical Security

Note: The *Basic Security Guide* contains a complete discussion of physical security on the AS/400 system.

___ The system unit and system console are in a secure location.

___ Backup media is protected from damage and theft.

___ The keylock switch setting on the processor unit is in the Secure or Auto position. The key is removed. The keys are kept separately, both under tight physical security. See the *Operator's Guide* for more information about the keylock switch.

___ Access to publicly located workstations and the console is restricted. Use the DSPOBJAUT command to see who has *CHANGE authority to the workstations. Look for AF entries in the audit journal with the object type field equal to *DEVD to find attempts to sign on at restricted workstations.

___ Sign-on for users with *ALLOBJ or *SERVICE special authority is limited to a few workstations. Check to see that the QLMTSECOFR system value is 1. Use the DSPOBJAUT command for devices to see if the QSECOFR profile has *CHANGE authority.

System Values

___ Security system values follow recommended guidelines. To print the security system values, type: `WRKSYSVAL *SEC OUTPUT(*PRINT)`. Two important system values to audit are:

- QSECURITY, which should be set to 30 or higher.
- QMAXSIGN, which should not be greater than 5.

Note: If the auditing function is active, an SV entry is written to the QAUDJRN journal whenever a system value is changed.

___ Decisions about system values are reviewed periodically, particularly when the system environment changes, such as the installation of new applications or a communications network.

IBM-Supplied User Profiles

___ The passwords have been changed for IBM-supplied user profiles. The following IBM-supplied user profiles are shipped with passwords equal to the user profile names:

QPGMR QSECOFR QSRV
QSRVBAS QSYSOPR QUSER

These passwords should be changed immediately after installing your system and periodically after installation. Verify that they have been changed by checking a DSPAUTUSR list for the date the passwords were changed and by attempting to sign on with the default passwords.

Note: See "IBM-Supplied User Profiles" on page 4-23 and Appendix B for more information about IBM-supplied user profiles.

___ The IBM passwords for dedicated service tools (DST) are changed. DST profiles do not appear on a DSPAUTUSR list. To verify that the passwords have been changed, start DST and attempt to use the default passwords. See the topic "Changing Passwords for Dedicated Service Tools (DST) Users" on page 4-23 for more information.

___ Signing on with IBM-supplied user profiles that are designed to be object owners is not permitted. Use a DSPAUTUSR list to verify that the following IBM-supplied user profiles have a password of *NONE:

QDBSHR QDFTOWN QDOC
QDSNX QFNC QGATE
QLPAUTO QLPINSTALL QSNADS
QSPL QSPLJOB QSYS
QTSTRQS

Password Control

___ Users can change their own passwords. Allowing users to define their own passwords reduces the need for users to write down their passwords. Users should have access to the CHGPWD command or to the Change Password function from the Operational Assistant menu.

___ A password change is required according to the organization's security guidelines, usually every 30 to 90 days. The QPWDEXPITV system value is set to meet the security guidelines.

___ If a user profile has a password expiration interval that is different from the system value, it meets the security guidelines. Review user profiles for a PWDEXPITV value other than *SYSVAL.

___ Trivial passwords are prevented by using the syster values to set the password rules and by using a password approval program. Use the WRKSYSVAl *SEC command and look at the settings for the values beginning with QPWD.

___ Group profiles have a password of *NONE. Check the DSPAUTUSR list.

User and Group Profiles

___ Each user is assigned a unique user profile. The QLMTDEVSSN system value should be set to 1. Although limiting each user to one device session a a time does not prevent sharing user profiles, it discourages it.

___ User profiles with *ALLOBJ special authority are limited, and are not used as group profiles. The DSPUSRPRF command can be used to check the special authorities for user profiles and to determine which profiles are group profiles. The topic "Printinç Selected User Profiles" on page 9-16 shows how tc use an output file and query to determine this.

___ The *Limit capabilities* field is *YES in the profiles of users who should be restricted to a set of menus. The topic "Printing Selected User Profiles" on page 9-16 gives an example of how to determine this.

___ Programmers are restricted from production librarie: Use the DSPOBJAUT command to determine the public and private authorities for production libraries and critical objects in the libraries.

"Planning Security for Programmers" on page 7-11 has more information about security and the programming environment.

___ Membership in a group profile is changed when job responsibilities change. To verify group membership, use one of these commands:

```
DSPAUTUSR SEQ(*GRPPRF)
DSPUSRPRF profile-name *GRPMBR
```

___ A naming convention is used for group profiles. Thi: allows them to be easily recognized when authoritie: are displayed.

___ The administration of user profiles is adequately organized. No user profiles have large numbers of private authorities. The topic "Examining Large Use Profiles" on page 9-16 discusses how to find and examine large user profiles on your system.

___ Employees are removed from the system immediately when they are transferred or released. Regularly review the DSPAUTUSR list to make sure onl active employees have access to the system. The

DO (Delete Object) entries in the audit journal can be reviewed to make sure user profiles are deleted immediately after employees leave.

___ Management regularly verifies the users authorized to the system. Use the DSPAUTUSR list.

___ The password for an inactive employee is set to *NONE. Use the DSPAUTUSR list to monitor the date each user last signed on the system.

___ Management regularly verifies the users with special authorities, particularly *ALLOBJ special authority. The topic "Printing Selected User Profiles" on page 9-16 gives an example of how to determine this.

Authorization Control

___ Owners of data understand their obligation to authorize users on a need-to-know basis.

___ Owners of objects regularly verify the authority to use the objects, including public authority. The WRKOBJOWN command provides a display for working with the authorities to all objects owned by a user profile.

___ Sensitive data is not public. Check the authority for user *PUBLIC for critical objects using the DSPOBJAUT command.

___ Authority to user profiles is controlled. The public authority to user profiles should be *EXCLUDE. This prevents users from submitting jobs that run under another user's profile.

___ Job descriptions are controlled:

- Job descriptions with public authority of *USE or greater are specified as USER(*RQD). This means jobs submitted using the job description must run using the submitter's profile.
- Job descriptions that specify a user have public authority *EXCLUDE. Authorization to use these job descriptions is controlled. This prevents unauthorized users from submitting jobs that run using another profile's authority.

To find out what job descriptions are on the system, type:

```
DSPOBJD OBJ(*ALL/*ALL) OBJTYPE(*JOBD) +
        DETAIL(*BASIC) OUTPUT(*PRINT)
```

To check the *User* parameter of a job description, use the Display Job Description (DSPJOBD) command. To check the authority to a job description, use the DSPOBJAUT command.

The CHKJOBDUSR command in QUSRTOOL can help you audit job descriptions.

Note: At security level 40 or 50, a user submitting a job using a job description that specifies a user profile name must have *USE authority to both the job description and the user profile. At all security levels, an attempt to submit or schedule a job without *USE authority to the user specified in the job description causes an AF entry with violation type J in the audit journal.

___ Users are not allowed to sign on by pressing the Enter key on the Sign On display. Make sure no workstation entries in subsystem descriptions specify a job description that has a user profile name specified for the USER parameter.

Default sign-on is prevented at security level 40 or 50, even if a subsystem description allows it. At all security levels, an AF entry with violation type S is written to the audit journal if default sign-on is attempted and a subsystem description is defined to allow it.

___ The library list in application programs is controlled to prevent a library that contains a similar program from being added before the production libraries. The topic "Security and Library Lists" on page 6-4 discusses methods for controlling the library list.

___ Programs that adopt authority are used only when required and are carefully controlled. See the topic "Analyzing Programs That Adopt Authority" on page 9-16 for an explanation of how to evaluate the use of the program adopt function.

___ Application program interfaces (APIs) are secured.

___ Good object security techniques are used to avoid performance problems.

Unauthorized Access

___ Security-related events are logged to the security auditing journal (QAUDJRN). The auditing function is active. The QAUDLVL system value specifies at least *AUTFAIL and *PGMFAIL. Regularly reviewing entries in the audit journal is the best method for detecting unauthorized attempts to access information.

___ The QMAXSIGN system value limits the number of consecutive incorrect access attempts to five or less. The QMAXSGNACN system value is set at 2 or 3.

___ The QSYSMSG message queue is created and monitored.

___ The audit journal is audited for repeated attempts by a user. (Authorization failures cause AF type entries in the audit journal.)

___ Programs fail that attempt to access objects using interfaces that are not supported. (QSECURITY system value is set to 40 or 50.)

___ User ID and password are required to sign on. Security levels 40 and 50 enforce this. At level 20 or 30, you must ensure that no subsystem descriptions have a workstation entry which uses a job description that has a user profile name.

Communications

___ Telephone communications is protected by call-back procedures.

___ Encryption is used on sensitive data.

___ Remote sign-on is controlled. The QRMTSIGN system value is set to *FRCSIGNON or a pass-through validation program is used.

___ Access to data from other systems, including personal computers, is controlled using the JOBACN, PCSACC, and DDMACC network attributes. The JOBACN network attribute should be *FILE.

Using the Security Audit Journal

The security audit journal is the primary source of auditing information on the system. A security auditor inside or outside your organization can use the auditing function provided by the system to gather information about security-related events that occur on the system.

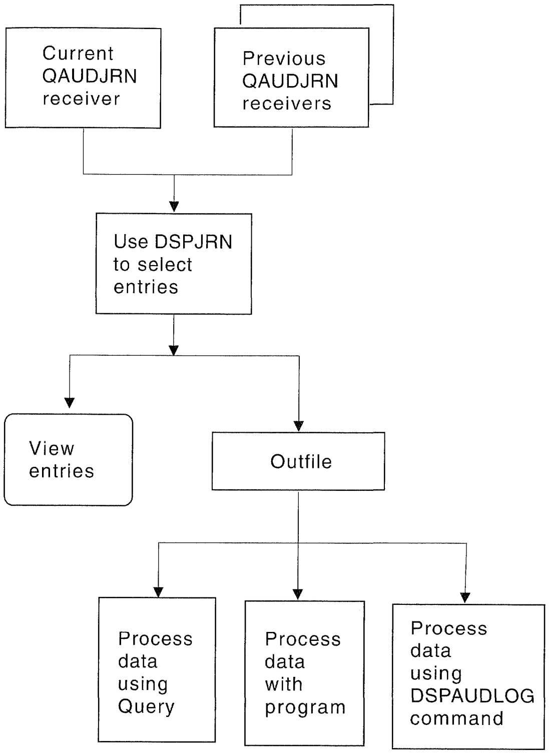
You can define auditing on your system at three different levels:

- System-wide auditing that occurs for all users.
- Auditing that occurs for specific objects.
- Auditing that occurs for specific users.

You use system values, user profile parameters, and object parameters to define auditing. "Planning Security Auditing" describes how to do this.

When a security-related event that may be audited occurs, the system checks whether you have selected that event for audit. If you have, the system writes a journal entry in the current receiver for the security auditing journal (QAUDJRN in library QSYS).

When you want to analyze the audit information you have collected in the QAUDJRN journal, you can use the Display Journal (DSPJRN) command. With this command, information from the QAUDJRN journal can be written to a database file. An application program or a query tool can be used to analyze the data. You can also use the DSPAUDLOG command from library QUSRTOOL to view audit journal information:


Current QAUDJRN receiver
Previous QAUDJRN receivers
Use DSPJRN to select entries
View entries
Outfile
Process data using Query
Process data with program
Process data using DSPAUDLOG command

RV2L260-0

Figure 9-1. Viewing QAUDJRN Information

The security auditing function is optional. You must take specific steps to set up security auditing.

The following sections describe how to plan, set up, and manage security auditing, what information is recorded, and how to view that information. Appendix F shows record layouts for the audit journal entries. Appendix G describes what operations are audited for each type of object.

Planning Security Auditing

To plan the use of security auditing on your system:

- Determine which security-relevant events you want to record for all system users. This is called action auditing.
- Check whether you need additional auditing for specific users.
- Decide whether you want to audit the use of specific objects on the system.
- Determine whether object auditing should be used for all users or specific users.

Planning the Auditing of Actions: The QAUDCTL (audit control) system value, the QAUDLVL (audit level) system value, and the AUDLVL (action auditing) parameter in user profiles work together to control action auditing:

- The QAUDLVL system value specifies which actions are audited for all users of the system
- The AUDLVL parameter in the user profile determines

which actions are audited for a specific user. The values for the AUDLVL parameter apply *in addition to* the values for the QAUDLVL system value.

- The QAUDCTL system value starts and stops action auditing.

Which events you choose to log depends on both your security objectives and your potential exposures. Table 9-1 on page 9-6 describes the possible audit level values and how you might use them. It shows whether they are available as a system value, a user profile parameter, or both. You can specify more than one value for both the QAUDLVL system value and the AUDLVL user profile parameter.

Table 9-2 on page 9-7 provides more information about the journal entries that are written for the action auditing values specified on the QAUDLVL system value and in the user profile. It shows:

- The type of entry written to the QAUDJRN journal.
- The field reference file that can be used to define the record when you create an output file with the DSPJRN command. Complete layouts for the field reference files are found in Appendix F.
- The detailed entry type. Some journal entry types are used to log more than one type of event. The detailed entry type field in the journal entry identifies the type of event.
- The ID of the message that can be used to define the entry-specific information in the journal entry. These messages are used by the DSPAUDLOG command in the QUSRTOOL library.

Table 9-1. Action Auditing Values

Possible Value	Available on QAUDLVL System Value	Available on CHGUSRAUD Command	Description
*NONE	Yes	Yes	If the QAUDLVL system value is *NONE, no actions are logged on a system-wide basis. Actions are logged for individual users based on the AUDLVL value in their user profiles. If the AUDLVL value in a user profile is *NONE, no additional action auditing is done for this user. Any actions specified for the QAUDLVL system value are logged for this user.
*AUTFAIL	Yes	No	**Authorization failures**: Unsuccessful attempts to sign on the system and to access objects are logged. *AUTFAIL can be used regularly to monitor users trying to perform unauthorized functions on the system. *AUTFAIL can also be used to assist with migration to a higher security level and to test resource security for a new application.
*CMD	No	Yes	**Commands**: The system logs command strings run by a user. If a command is run from a CL program that is created with LOG(*NO) and ALWRTVSRC(*NO), only the command name and library name are logged. *CMD may be used to record the actions of a particular user, such as the security officer.
*CREATE	Yes	Yes	**Creating objects**: The system writes a journal entry when a new or replacement object i created. *CREATE may be used to monitor when programs are created or recompiled.
*DELETE	Yes	Yes	**Deleting objects**: The system writes a journal entry when an object is deleted.
*JOBDTA	Yes	Yes	**Job tasks**: Actions that affect a job are logged, such as starting or stopping the job, holding, releasing, canceling, or changing it. *JOBDTA may be used to monitor who is running batch jobs.
*OBJMGT	Yes	Yes	**Object management tasks**: Moving an object to a different library or renaming it is logged. *OBJMGT may be used to detect copying confidential information by moving the object to a different library.
*OFCSRV	Yes	Yes	**OfficeVision/400 tasks**: Changing the system distribution directory and opening a mail log are recorded. Actions performed on specific items in the mail log are not recorded. *OFCSRV may be used to detect attempts to change how mail is routed or to monitor opening another user's mail log.
*PGMADP	Yes	Yes	**Adopting authority**: The system writes a journal entry when adopted authority is used tc gain access to an object. *PGMADP may be used to test where and how a new application uses adopted authority.
*PGMFAIL	Yes	No	**Program failures**: The system writes a journal entry when a program causes an integrity error. *PGMFAIL may be used to assist with migration to a higher security level or to test a new application.
*PRTDTA	Yes	No	**Printing functions**: Printing a spooled file or printing directly from a program is logged. *PRTDTA may be used to detect printing confidential information.
*SAVRST	Yes	Yes	**Save and restore operations**: Saving objects from the system and restoring objects to the system is logged. *SAVRST may be used to detect attempts to restore unauthorized objects or to track attempts to save objects.
*SECURITY	Yes	Yes	**Security tasks**: Security-relevant events, such as changing a user profile or system value, are logged. *SECURITY may be used to keep a record of all security activity.
*SERVICE	Yes	Yes	**Service tasks**: The use of service tools, such as DMPOBJ (Dump Object) and STRCPYSCN (Start Copy Screen), is logged. *SERVICE may be used to detect attempts to circumvent security by using service tools.
*SPLFDTA	Yes	Yes	**Operations on spooled files**: Actions performed on spooled files are logged, including creating, copying, and sending. *SPLFDTA may be used to detect attempts to print or send confidential data.
*SYSMGT	Yes	Yes	**System management tasks**: The system writes a journal entry for system management activities, such as changing a reply list or the power on/off schedule. *SYSMGT may be used to detect attempts to use system management functions to circumvent security controls.

Table 9-2 (Page 1 of 2). Security Auditing Journal Entries

Action or Object Auditing Value	Journal Entry Type	Field Reference File	Detailed Entry	Message ID	Description
Action Auditing:					
*AUTFAIL [1]	AF	QASYAFJE	A	CPI2246	Attempt made to access an object or perform an operation to which the user was not authorized.
			J	CPI2248	Attempt made to submit or schedule a job under a job description which has a user profile specified. The submitter did not have *USE authority to the user profile.
			P	CPI2270	Attempt made to use a profile handle that is not valid on the QWTSETP API.
			S	CPI2249	Attempt made to sign on without entering a user ID or a password.
			U	CPI2296	A user permission request was not valid.
	PW	QASYPWJE	P	CPI2251	An incorrect password was entered
			U	CPI2252	An incorrect user ID was entered
			A	CPI2292	An APPC bind failed.
*CMD [2]	CD	QASYCDJE	C	CPI2286	A command was run.
			L	CPI2286	An S/36E control language statement was run.
			O	CPI2286	An S/36E operator control command was run.
			P	CPI2286	An S/36E procedure was run.
			U	CPI2286	An S/36E utility control statement was run.
*CREATE [3]	CO	QASYCOJE	N	CPI2277	Creation of a new object, except creation of objects in QTEMP library.
			R	CPI2278	Creation of an object that replaces an existing object.
*DELETE [3]	DO	QASYDOJE	A	CPI2263	All delete operations, except deleting objects from QTEMP library.
*JOBDTA	JS	QASYJSJE	A	CPI2288	The ENDJOBABN command was used.
			B	CPI2288	A job was submitted.
			C	CPI2288	A job was changed.
			E	CPI2288	A job was ended.
			H	CPI2288	A job was held.
			I	CPI2288	A job was disconnected.
			N	CPI2288	The ENDJOB command was used.
			P	CPI2288	A program start request was attached to a prestart job.
			R	CPI2288	A held job was released.
			S	CPI2288	A job was started.
*OBJMGT [3]	OM	QASYOMJE	M	CPI2281	An object was moved to a different library.
			R	CPI2282	An object was renamed.
*OFCSRV	ML	QASYMLJE	O	CPI2289	A mail log was opened.
	SD	QASYSDJE	S	CPI2293	A change was made to the system distribution directory.
*PGMADP	AP	QASYAPJE	S	CPI2287	A program started that adopts owner authority. The start entry is written the first time adopted authority is used to gain access to an object, not when the program enters the program stack.
			E	CPI2287	A program ended that adopts owner authority. The end entry is written when the program leaves the program stack. If the same program occurs more than once in the program stack, the end entry is written when the highest (last) occurrence of the program leaves the stack.
*PGMFAIL [1]	AF	QASYAFJE	B	CPI2268	A program ran a restricted machine interface instruction.
			C	CPI2250	A program which failed the restore-time program validation checks was restored. Information about the failure is in the *Validation Value Violation Type* field of the record.
			D	CPI2247	A program accessed an object through an unsupported interface or callable program not listed as a callable API.
			R	CPI2274	Attempt made to update an object that is defined as read-only. (Enhanced hardware storage protection is logged only at security level 40 and higher)
*PRTDTA [1]	PO	QASYPOJE	D	CPI2290	Printer output was printed directly to a printer.
			S	CPI2290	Printer output was spooled and printed.
*SAVRST [3]	OR	QASYORJE	N	CPI2279	A new object was restored to the system.
			E	CPI2280	An object was restored that replaces an existing object.
	RA	QASYRAJE	A	CPI2261	The system changed the authority to an object being restored. [4]

Table 9-2 (Page 2 of 2). Security Auditing Journal Entries

Action or Object Auditing Value	Journal Entry Type	Field Reference File	Detailed Entry	Message ID	Description
	RJ	QASYRJJE	A	CPI2259	A job description that contains a user profile name was restored.
	RO	QASYROJE	A	CPI2260	The object owner was changed to QDFTOWN during restore operation.[4]
	RP	QASYRPJE	A	CPI2258	A program that adopts owner authority was restored.
	RU	QASYRUJE	A	CPI2262	Authority was restored for a user profile using the RSTAUT command.
*SECURITY	AD	QASYADJE	D	CPI2285	Auditing of a DLO was changed with CHGDLOAUD command.
			O	CPI2285	Auditing of an object was changed with CHGOBJAUD command.
			U	CPI2285	Auditing for a user was changed with CHGUSRAUD command.
	CA	QASYCAJE	A	CPI2253	Changes to authorization list or object authority.
	CP	QASYCPJE	A	CPI2266	Create, change, or restore operation of user profile.
	DS	QASYDSJE	A	CPI2267	Request to reset DST QSECOFR password to system-supplied default.
	JD	QASYJDJE	A	CPI2264	The USER parameter of a job description was changed.
	NA	QASYNAJE	A	CPI2257	A network attribute was changed.
	OW	QASYOWJE	A	CPI2254	Object ownership was changed.
	PA	QASYPAJE	A	CPI2255	A program was changed to adopt owner authority.
	PS	QASYPSJE	A	CPI2273	A target user profile was changed during a pass-through session.
			E	CPI2291	An office user ended work on behalf of another user.
			H	CPI2272	A profile handle was generated through the QSYGETPH API.
			S	CPI2291	An office user started work on behalf of another user.
	SE	QASYSEJE	A	CPI2265	A subsystem routing entry was changed.
	SV	QASYSVJE	A	CPI2256	A system value was changed.
*SERVICE	ST	QASYSTJE	A	CPI228F	A service tool was used.
*SPLFDTA	SF	QASYSFJE	A	CPI2294	A spooled file was read by someone other than the owner.
			C	CPI2294	A spooled file was created.
			D	CPI2294	A spooled file was deleted.
			H	CPI2294	A spooled file was held.
			I	CPI2294	An inline file was created.
			R	CPI2294	A spooled file was released.
			U	CPI2294	A spooled file was changed.
*SYSMGT	SM	QASYSMJE	B	CPI228E	Backup options were changed using Operational Assistant.
			C	CPI228E	Automatic cleanup options were changed using Operational Assistant.
			D	CPI228E	A DRDA change was made.
			F	CPI228E	An HFS file system was changed.
			N	CPI228E	A network file operation was performed.
			O	CPI228E	A backup list was changed using Operational Assistant.
			P	CPI228E	The power on/off schedule was changed using Operational Assistant.
			S	CPI228E	The system reply list was changed.
Object Auditing:					
*CHANGE	YC	QASYYCJE	C	CPI228A	A document library object was changed.
	ZC	QASYZCJE	C	CPI228C	An object was changed.
*ALL [5]	YR	QASYYRJE	R	CPI228B	A document library object was read.
	ZR	QASYZRJE	R	CPI228D	An object was read.

[1] This value can only be specified for the QAUDLVL system value. It is not a value for the AUDLVL parameter of a user profile.

[2] This value can only be specified for the AUDLVL parameter of a user profile. It is not a value for the QAUDLVL system value.

[3] If object auditing is active for an object, an audit record is written for a create, delete, object management, or restore operation even if these actions are not included in the audit level.

[4] See the topic “Restoring Objects” on page 8-2 for information about authority changes which may occur when an object is restored.

[5] When *ALL is specified, the entries for both *CHANGE and *ALL (YC, YR, ZC, and ZR) are written.

Planning the Auditing of Object Access: The QAUDCTL system value, the OBJAUD value for an object, and the OBJAUD value for a user profile work together to control object auditing. The OBJAUD value for the object and the OBJAUD value for the user who is using the object determine whether a specific access should be logged. The QAUDCTL system value starts and stops the object auditing function.

Table 9-3 shows how the OBJAUD values for the object and the user profile work together.

Table 9-3. How Object and User Auditing Work Together

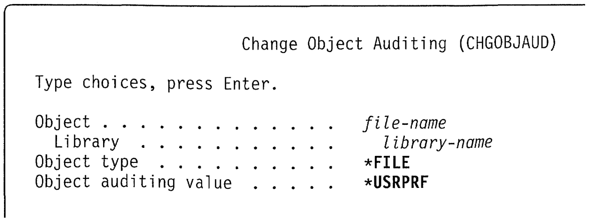
OBJAUD Value for Object	**OBJAUD Value for User**		
	*NONE	*CHANGE	*ALL
*NONE	None	None	None
*USRPRF	None	Change	Change and Use
*CHANGE	Change	Change	Change
*ALL	Change and Use	Change and Use	Change and Use

You can use object auditing to keep track of all users accessing a critical object on the system. You can also use object auditing to keep track of all the object accesses by a particular user. Object auditing is a flexible tool that allows you to monitor those object accesses that are important to your organization.

Taking advantage of the capabilities of object auditing requires careful planning. Poorly designed auditing may generate many more audit records than you can analyze, and can have a severe impact on system performance. For example, setting the OBJAUD value to *ALL for a library results in an audit entry being written every time the system searches for an object in that library. For a heavily used library on a busy system, this would generate a very large number of audit journal entries.

The following are some examples of how to use object auditing.

- If certain critical files are used throughout your organization, you may periodically review who is accessing them using a sampling technique:
  1. Set the OBJAUD value for each critical file to *USRPRF using the Change Object Auditing command:

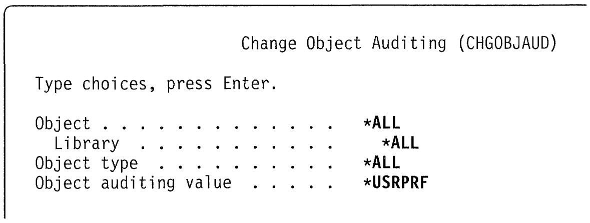


```
                 Change Object Auditing (CHGOBJAUD)

Type choices, press Enter.

Object . . . . . . . . . . . . .   file-name
  Library  . . . . . . . . . . .     library-name
Object type  . . . . . . . . . .   *FILE
Object auditing value  . . . . .   *USRPRF
```

  2. Set the OBJAUD value for each user in your sample to *CHANGE or *ALL using the CHGUSRAUD command.
  3. Make sure the QAUDCTL system value includes *OBJAUD.
  4. When sufficient time has elapsed to collect a representative sample, set the OBJAUD value in the user profiles to *NONE or remove *OBJAUD from the QAUDCTL system value.
  5. Analyze the audit journal entries using the techniques described in "Analyzing Audit Journal Entries with Query or a Program" on page 9-13.
- If you are concerned about who is using a particular file, you can collect information about all accesses of that file for a period of time:
  1. Set object auditing for the file independent of user profile values:

```
CHGOBJAUD OBJECT(library-name/file-name)
          OBJTYPE(*FILE)  OBJAUD(*CHANGE or *ALL)
```

  2. Make sure the QAUDCTL system value includes *OBJAUD.
  3. When sufficient time has elapsed to collect a representative sample, set the OBJAUD value in the object to *NONE.
  4. Analyze the audit journal entries using the techniques described in "Analyzing Audit Journal Entries with Query or a Program" on page 9-13.
- To audit all object accesses for a specific user, do the following:
  1. Set the OBJAUD value for all objects to *USRPRF using the CHGOBJAUD command:



```
                 Change Object Auditing (CHGOBJAUD)

Type choices, press Enter.

Object . . . . . . . . . . . . .   *ALL
  Library  . . . . . . . . . . .     *ALL
Object type  . . . . . . . . . .   *ALL
Object auditing value  . . . . .   *USRPRF
```

     Warning: Depending on how many objects are on your system, this command may take many hours to run. Setting up object auditing for all objects on the system is usually not necessary and will severely degrade performance. Selecting a subset of object types and libraries for auditing is recommended.
  2. Set the OBJAUD value for the specific user profile to *CHANGE or *ALL using the CHGUSRAUD command.
  3. Make sure the QAUDCTL system value includes *OBJAUD.
  4. When you have collected a specific sample, set the OBJAUD value for the user profile to *NONE.

Displaying Object Auditing: Use the DSPOBJD command to display the current object auditing level for an object. Use the DSPDLOAUD command to display the current object auditing level for a document library object.

Setting Default Auditing for Objects: You can use the QCRTOBJAUD system value and the CRTOBJAUD value for libraries to set object auditing for new objects that are created. For example, if you want all new objects in the INVLIB library to have an audit value of *USRPRF, use the following command:

```
CHGLIB LIB(INVLIB) CRTOBJAUD(*USRPRF)
```

This command affects the auditing value of new objects only. It does not change the auditing value of objects that already exist in the library.

Use the default auditing values carefully. Improper use could result in many unwanted entries in the security audit journal. Effective use of the object auditing capabilities of the system requires careful planning.

Preventing Loss of Auditing Information:

Two system values control what the system does when error conditions may cause the loss of audit journal entries.

Audit Force Level: The QAUDFRCLVL system value determines how often the system writes audit journal entries from memory to auxiliary storage. The QAUDFRCLVL system value works like the force level for database files. You should follow similar guidelines in determining the correct force level for your installation.

If you allow the system to determine when to write entries to auxiliary storage, it balances the performance impact against the potential loss of information in a power outage. *SYS is the default and the recommended choice.

If you set the force level to a low number, you minimize the possibility of losing audit records, but you may notice a negative performance impact. If your installation requires that no audit records be lost in a power failure, you must set the QAUDFRCLVL to 1.

Audit End Action: The QAUDENDACN system value determines what the system does if it is unable to write an entry to the audit journal. The default value is *NOTIFY. The system does the following if it is unable to write audit journal entries and QAUDENDACN is *NOTIFY:

1. The QAUDCTL system value is set to *NONE to prevent additional attempts to write entries.
2. Message CPI2283 is sent to the QSYSOPR message queue and the QSYSMSG message queue (if it exists) every hour until auditing is successfully restarted.
3. Normal processing continues.
4. If an IPL is performed on the system, message CPI2284 is sent to the QSYSOPR and QSYSMSG message queues during the IPL.

   Note: In most cases, performing an IPL resolves the problem that caused auditing to fail. After you have restarted your system, set the QAUDCTL system value to the correct value. The system attempts to write an audit journal record whenever this system value is changed.

You can set the QAUDENDACN to power down your system if auditing fails (*PWRDWNSYS). Use this value only if your installation requires that auditing be active for the system to run. If the system is unable to write an audit journal entry and the QAUDENDACN system value is *PWRDWNSYS, the following happens:

1. The system powers down immediately (the equivalent of issuing the PWRDWNSYS *IMMED command).
2. SRC code B900 3D10 is displayed.

Next, you must do the following:

1. Start an IPL from the system unit. Make sure that the device specified in the system console (QCONSOLE) system value is powered on.
2. To complete the IPL, a user with *ALLOBJ and *AUDIT special authority must sign on at the console.
3. The system starts in a restricted state with a message indicating that an auditing error caused the system to stop.
4. The QAUDCTL system value is set to *NONE.
5. To restore the system to normal, set the QAUDCTL system value to a value other than none. When you change the QAUDCTL system value, the system attempts to write an audit journal entry. If it is successful, the system returns to a normal state.

   If the system does not successfully return to a normal state, use the job log to determine why auditing has failed. Correct the problem and attempt to reset the QAUDCTL value again.

Setting up Security Auditing

Overview

Purpose:	Set up the system to collect security events in the QAUDJRN journal.
How To:	CRTJRNRCV CRTJRN QSYS/QAUDJRN WRKSYSVAL *SEC CHGOBJAUD CHGDLOAUD CHGUSRAUD
Authority:	*ADD authority to QSYS and to journal receiver library *AUDIT special authority
Journal Entry:	CO (create object) SV (system value change) AD (object and user audit changes)
Notes:	QSYS/QAUDJRN must exist before QAUDCTL can be changed.

To set up security auditing, do the following steps. Setting up auditing requires *AUDIT special authority.

1. Create a journal receiver in a library of your choice by using the Create Journal Receiver (CRTJRNRCV) command. This example uses a library called JRNLIB for journal receivers.

   ```
   CRTJRNRCV  JRNRCV(JRNLIB/AUDRCV0001) +
              AUT(*EXCLUDE)   +
              TEXT('Auditing Journal Receiver')
   ```

   Place the journal receiver in a library that is saved regularly. Choose a journal receiver name which can be used to create a naming convention for future journal receivers, such as AUDRCV0001. You can use the *GEN option when you change journal receivers to continue the naming convention. Specify *EXCLUDE on the AUT parameter to limit access to the information stored in the journal.

2. Create the QSYS/QAUDJRN journal by using the Create Journal (CRTJRN) command:

   ```
   CRTJRN  JRN(QSYS/QAUDJRN) +
           JRNRCV(JRNLIB/AUDRCV0001) +
           AUT(*EXCLUDE) TEXT('Auditing Journal')
   ```

   The name QSYS/QAUDJRN *must* be used. Specify the name of the journal receiver you created in the previous step. Specify *EXCLUDE on the AUT parameter to limit access to the information stored in the journal. You must have authority to add objects to QSYS to create the journal.

   The *Advanced Backup and Recovery Guide* provides more information about working with journals and journal receivers.

3. Set the audit level (QAUDLVL) system value using the WRKSYSVAL command. The QAUDLVL system value determines which actions are logged to the audit journal for all users on the system. See “Planning the Auditing of Actions” on page 9-4.

4. Set action auditing for individual users if necessary using the CHGUSRAUD command. See “Planning the Auditing of Actions” on page 9-4.

5. Set object auditing for specific objects if necessary using the CHGOBJAUD and CHGDLOAUD commands. See “Planning the Auditing of Object Access” on page 9-9.

6. Set object auditing for specific users if necessary using the CHGUSRAUD command.

7. Set the QAUDENDACN system value to control what happens if the system cannot access the audit journal. See “Audit End Action” on page 9-10.

8. Set the QAUDFRCLVL system value to control how often audit records are written to auxiliary storage. See “Preventing Loss of Auditing Information” on page 9-10.

9. Start auditing by setting the QAUDCTL system value to a value other than *NONE.

   The QSYS/QAUDJRN journal must exist before you can change the QAUDCTL system value to a value other than *NONE. When you start auditing, the system attempts to write a record to the audit journal. If the attempt is not successful, you receive a message and auditing does not start.

Managing the Audit Journal and Journal Receivers

The auditing journal, QSYS/QAUDJRN, is intended solely for security auditing. Files should not be journaled to the audit journal. User entries should not be sent to this journal using the Send Journal Entry (SNDJRNE) command. Special locking protection is used to ensure that the system can write audit entries to the audit journal.

When auditing is active (the QAUDCTL system value is not *NONE), the system arbitrator job (QSYSARB) holds a lock on the QSYS/QAUDJRN journal. You cannot perform certain operations on the audit journal when auditing is active, such as:

- DLTJRN command
- ENDJRNAP command
- ENDJRNPF command
- APYJRNCHG command
- RMVJRNCHG command
- DMPOBJ or DMPSYSOBJ command
- Move operations
- Restoring the journal
- Operations that work with authority, such as the GRTOBJAUT command
- WRKJRN command

The information recorded in the security journal entries is described in Appendix F. All security entries in the audit journal have a journal code of T. In addition to security entries, system entries also appear in the journal QAUDJRN. These are entries with a journal code of J, which relate to initial program load (IPL) and general operations performed on journal receivers (for example, saving the receiver).

If damage occurs to the journal or to its current receiver so that the auditing entries cannot be journaled, the QAUDENDACN system value determines what action the system takes. Recovery from a damaged journal or journal receiver is the same as for other journals.

If the journal receiver reaches a storage threshold, a message is sent to the threshold message queue specified for the journal. The messages indicates that the receiver has reached its threshold and some action must be taken. If this occurs, the Change Journal (CHGJRN) command must be used to detach the receiver and attach a new one.

The default message queue for a journal is QSYSOPR. If your installation has a large volume of messages in the QSYSOPR message queue, you may want to associate a different message queue, such AUDMSG, with the QAUDJRN journal. You can use a message handling program to monitor the AUDMSG message queue. When a journal threshold warning is received, you can automatically attach a new receiver.

Warning: The automatic cleanup function provided using Operational Assistant menus does not clean up the QAUDJRN receivers. You should regularly detach, save, and delete QAUDJRN receivers to avoid problems with disk space.

See the *Advanced Backup and Recovery Guide* for complete information about managing journals and journal receivers.

Note: The QAUDJRN journal is created during an IPL if it does not exist and the QAUDCTL system value is set to a value other than *NONE. This occurs only after an unusual situation, such as replacing a disk device or clearing an auxiliary storage pool.

Saving and Deleting Audit Journal Receivers

Overview

Purpose:	To assign a new audit journal receiver; To save and delete the old receiver
How To:	CHGJRN QSYS/QAUDJRN JRNRCV(*GEN) SAVOBJ (to save old receiver) DLTJRNRCV (to delete old receiver)
Authority:	*ALL authority to journal receiver *USE authority to journal
Journal Entry:	J (system entry to QAUDJRN)
Notes:	Select a time when the system is not busy.

You should regularly detach the current audit journal receiver and attach a new one for two reasons:

- Analyzing journal entries is easier if each journal receiver contains the entries for a specific, manageable time period.
- Large journal receivers can affect system performance, in addition to taking valuable space on auxiliary storage.

If you have set up action auditing and object auditing to log many different events, you may need to change journal receivers daily. If you log only a few events, you may want to change receivers to correspond with the backup schedule for the library containing the journal receiver.

You use the CHGJRN command to detach a receiver and attach a new receiver. You should do this at a time when the system is not at maximum use to avoid affecting performance.

Use the following procedure to detach, save and delete a journal receiver:

1. Type `CHGJRN JRN(QAUDJRN) JRNRCV(*GEN)`. This command:
   a. Detaches the currently attached receiver.
   b. Creates a new receiver with the next sequential number.
   c. Attaches the new receiver to the journal.

   For example, if the current receiver is AUDRCV0003, th system creates and attaches a new receiver called AUDRCV0004.

   The Work with Journal Attributes (WRKJRNA) commanc tells you which receiver is currently attached: `WRKJRNA QAUDJRN`. The Work with Journals command gives you complete information about the journal: `WRKJRN QAUDJRN`
2. Use the Save Object (SAVOBJ) command to save the journal receiver. Specify object type *JRNRCV.
3. Use the Delete Journal Receiver (DLTJRNRCV) command to delete the receiver. If you try to delete the receiver without saving it, you receive a warning message.

Stopping the Audit Function

You may want to use the audit function periodically, rather than all the time. For example, you might want to use it when testing a new application. Or you might use it to perform a quarterly security audit.

To stop the auditing function, do the following:

1. Use the WRKSYSVAL command to change the QAUDCTL system value to *NONE. This stops the system from logging any more security events.
2. Detach the current journal receiver using the CHGJRN command.
3. Save and delete the receiver, using the SAVOBJ and DLTJRNRCV commands.
4. You can delete the QAUDJRN journal once you change QAUDCTL to *NONE. However, if you plan to resume security auditing in the future, you may want to leave the QAUDJRN journal on the system.

Analyzing Audit Journal Entries

Once you have set up the security auditing function, you can use several different methods to analyze the events that are logged:

- Viewing selected entries at your workstation
- Using a query tool or program to analyze entries
- Using the Display Audit Log (DSPAUDLOG) command

You can also use the Receive Journal Entry (RCVJRNE) command on the QAUDJRN journal to receive the entries as they are written to the QAUDJRN journal.

Viewing Audit Journal Entries

Overview

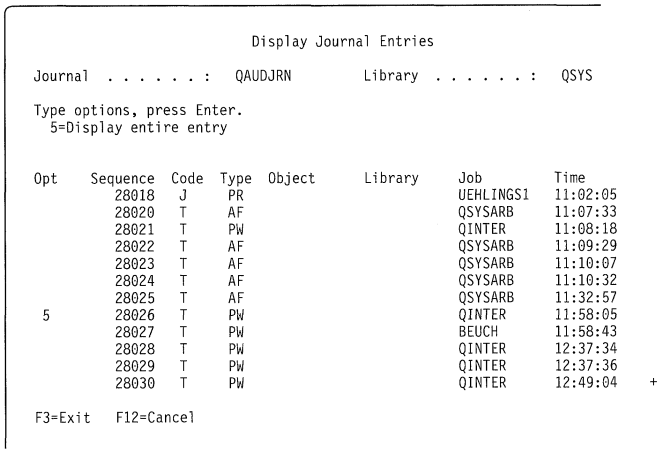
Purpose: View QAUDJRN entries

How To: DSPJRN (Display Journal command)

Authority: *USE authority to QSYS/QAUDJRN
*USE authority to journal receiver

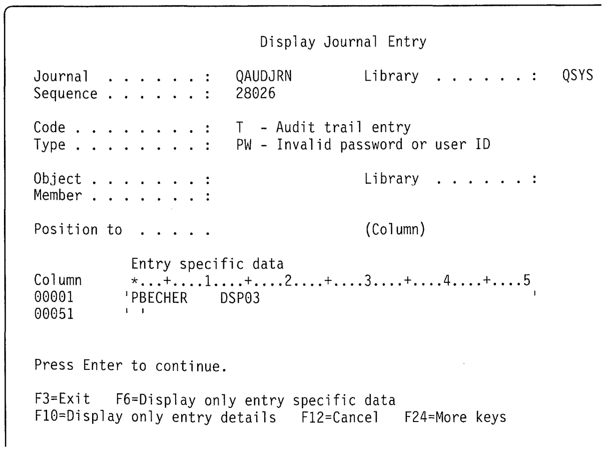
The Display Journal (DSPJRN) command allows you to view selected journal entries at your workstation. To view journal entries, do the following:

1. Type DSPJRN QAUDJRN and press F4. On the prompt display, you can enter information to select the range of entries that is shown. For example, you can select all entries in a specific range of dates, or you can select only a certain type of entry, such as an incorrect sign-on attempt (journal entry type PW).
2. When you press the Enter key, you see the Display Journal Entries display:



```
                        Display Journal Entries

Journal . . . . . . :   QAUDJRN        Library . . . . . . :   QSYS

Type options, press Enter.
  5=Display entire entry

Opt    Sequence  Code  Type  Object      Library      Job          Time
          28018   J     PR                            UEHLINGS1    11:02:05
          28020   T     AF                            QSYSARB      11:07:33
          28021   T     PW                            QINTER       11:08:18
          28022   T     AF                            QSYSARB      11:09:29
          28023   T     AF                            QSYSARB      11:10:07
          28024   T     AF                            QSYSARB      11:10:32
          28025   T     AF                            QSYSARB      11:32:57
 5        28026   T     PW                            QINTER       11:58:05
          28027   T     PW                            BEUCH        11:58:43
          28028   T     PW                            QINTER       12:37:34
          28029   T     PW                            QINTER       12:37:36
          28030   T     PW                            QINTER       12:49:04     +

F3=Exit   F12=Cancel
```

3. Use option 5 (Display entire entry) to see information about a specific entry:



```
                        Display Journal Entry

Journal . . . . . . :   QAUDJRN        Library . . . . . . :   QSYS
Sequence . . . . . . :  28026

Code . . . . . . . . :  T  - Audit trail entry
Type . . . . . . . . :  PW - Invalid password or user ID

Object . . . . . . . :                 Library . . . . . . :
Member . . . . . . . :

Position to  . . . . .                 (Column)

            Entry specific data
Column      *...+....1....+....2....+....3....+....4....+....5
00001       'PBECHER    DSP03                                    '
00051       ' '

Press Enter to continue.

F3=Exit   F6=Display only entry specific data
F10=Display only entry details   F12=Cancel   F24=More keys
```

4. You can use F6 (Display only entry specific data) for entries with a large amount of entry-specific data. You can also select a hexadecimal version of that display. You can use F10 to display details about the journal entry without any entry-specific information.

   Appendix F contains the layout for each type of QAUDJRN journal entry.

Analyzing Audit Journal Entries with Query or a Program

Overview

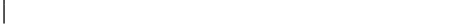
Purpose: Display or print selected information from journal entries.

How To: DSPJRN OUTPUT(*OUTFILE)
Create query or program
Run query or program

Authority: *USE authority to QSYS/QAUDJRN
*USE authority to journal receiver
*ADD authority to library for output file

You can use the Display Journal (DSPJRN) command to write selected entries from the audit journal receivers to an output file. You can use a program or a query to view the information in the output file.

For the output parameter of the DSPJRN command, specify *OUTFILE. You see an additional display prompting you for information about the output file:



```
                        Display Journal (DSPJRN)

Type choices, press Enter.

Output . . . . . . . . . . . . . > *OUTFILE
Outfile format . . . . . . . . .   *TYPE2
File to receive output . . . . .   dspjrnout
  Library  . . . . . . . . . . .     mylib
Output member options:
  Member to receive output . . .   *FIRST
  Replace or add records . . . .   *REPLACE
Entry data length:
  Field data format  . . . . . .   *OUTFILFMT
  Variable length field length
  Allocated length . . . . . . .
```

All security-related entries in the audit journal contain the same heading information, which includes the entry type, the date of the entry, and the job that caused the entry. A record format (QJORDJE2) is provided to define these fields when you specify *TYPE2 as the outfile format parameter.

In the QJORDJE2 record format, all of the entry-specific data (after position 156 in the record) is combined in a single field. Table F-1 on page F-2 shows the layout of this record format. Use the QJORDJE2 format if you want to analyze multiple entry types and view only summary information.

If you want to perform a detailed analysis of a particular entry type, use one of the field reference files provided. For example, to create an output file called AUDJRNAF in QGPL that includes only authority failure entries:

1. Create an empty output file with the format defined for AF journal entries:

   ```
   CRTDUPOBJ OBJ(QASYAFJE) FROMLIB(QSYS) +
     OBJTYPE(*FILE) TOLIB(QGPL) NEWOBJ(AUDJRNAF)
   ```

2. Use the DSPJRN command to write selected journal entries to the output file:

   ```
   DSPJRN JRN(QAUDJRN) ... +
      JRNCDE(T) ENTTYP(AF) OUTPUT(*OUTFILE) +
      OUTFILFMT(*TYPE2) OUTFILE(QGPL/AUDJRNAF)
   ```

3. Use Query or a program to analyze the information in the AUDJRNAF file.

Table 9-2 on page 9-7 shows the name of the field reference file for each entry type. Appendix F shows the file layouts for each field reference file.

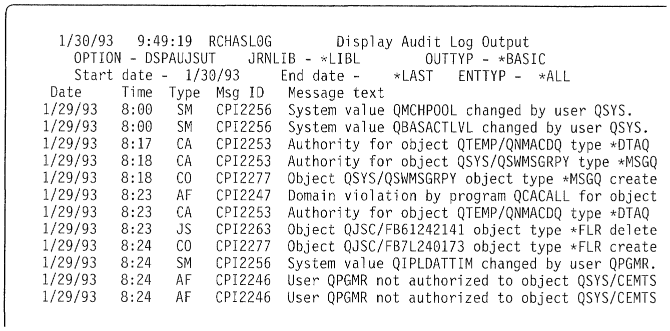
Following are a few examples of how you might use QAUDJRN information:

- If you suspect someone is trying to break into your system:
  1. Make sure the QAUDLVL system value includes *AUTFAIL.
  2. Use the CRTDUPOBJ object command to create an empty output file with the QASYAFJE format.
  3. A PW type journal entry is logged when someone enters an incorrect user ID or password on the Sign On display. Use the DSPJRN command to write PW type journal entries to the output file.
  4. Create a query program that displays or prints the date, time, and workstation for each journal entry. This information should help you determine where and when the attempts are occurring.
- If you want to test the resource security you have defined for a new application:
  1. Make sure the QAUDLVL system value includes *AUTFAIL.
  2. Run application tests with different user IDs.
  3. Use the CRTDUPOBJ object command to create an empty output file with the QASYAFJE format.
  4. Use the DSPJRN command to write AF type journal entries to the output file.
  5. Create a query program that displays or prints information about the object, job and user. This information should help you determine users and application functions are causing authority failures.
- If you are planning a migration to security level 40:
  1. Make sure the QAUDLVL system value includes *PGMFAIL and *AUTFAIL.
  2. Use the CRTDUPOBJ object command to create an empty output file with the QASYAFJE format.
  3. Use the DSPJRN command to write AF type journal entries to the output file.
  4. Create a query program that selects the type of violations you are experiencing during your test and prints information about the job and program that causes each entry.

Note: Table 9-2 on page 9-7 shows which journal entry is written for each authority violation message.

Using DSPAUDLOG to Analyze Audit Journal Entries: When you view journal entries using the DSPJRN command, they are difficult to understand, because the information is not separated into fields. The DSPAUDLOG tool in the QUSRTOOL library can be used to print descriptive text for entries in the QAUDJRN journal.

The following display shows sample output from the DSPAUDLOG command:



```
  1/30/93   9:49:19  RCHASLOG        Display Audit Log Output
    OPTION - DSPAUJSUT    JRNLIB - *LIBL        OUTTYP - *BASIC
    Start date -  1/30/93     End date -    *LAST   ENTTYP -  *ALL
 Date      Time  Type  Msg ID   Message text
1/29/93    8:00   SM   CPI2256  System value QMCHPOOL changed by user QSYS.
1/29/93    8:00   SM   CPI2256  System value QBASACTLVL changed by user QSYS.
1/29/93    8:17   CA   CPI2253  Authority for object QTEMP/QNMACDQ type *DTAQ
1/29/93    8:18   CA   CPI2253  Authority for object QSYS/QSWMSGRPY type *MSGQ
1/29/93    8:18   CO   CPI2277  Object QSYS/QSWMSGRPY object type *MSGQ create
1/29/93    8:23   AF   CPI2247  Domain violation by program QCACALL for object
1/29/93    8:23   CA   CPI2253  Authority for object QTEMP/QNMACDQ type *DTAQ
1/29/93    8:23   JS   CPI2263  Object QJSC/FB61242141 object type *FLR delete
1/29/93    8:24   CO   CPI2277  Object QJSC/FB7L240173 object type *FLR create
1/29/93    8:24   SM   CPI2256  System value QIPLDATTIM changed by user QPGMR.
1/29/93    8:24   AF   CPI2246  User QPGMR not authorized to object QSYS/CEMTS
1/29/93    8:24   AF   CPI2246  User QPGMR not authorized to object QSYS/CEMTS
```

Note: The printed version shows the entire text of the messages.

DSPAUDLOG displays the text of the CPIxxxx message associated with the journal entry. The message defines entry-specific data into fields and is easier to understand than the DSPJRN display. Table 9-2 on page 9-7 shows which messages are associated with each journal entry type.

Output from the DSPAUDLOG command can be directed either to a workstation or to a printer. The printed version can include both the message text and message help. DSPAUDLOG output is useful for:

- Reviewing messages in the security journal
- Printing a history of security violations

Other Techniques for Monitoring Security

The security audit journal (QAUDJRN) is the primary source of information about security-related events on your system. The following sections discuss other ways to observe security-related events and the security values on your system.

Monitoring Security Messages

Some security-relevant events, such as incorrect sign-on attempts, cause a message in the QSYSOPR message queue. You can also create a separate message queue called QSYSMSG in the QSYS library.

If you create the QSYSMSG message queue in the QSYS library, messages about critical system events are sent to that message queue as well as to QSYSOPR. The QSYSMSG message queue can be monitored separately by a program or a system operator. This provides additional protection of your system resources. Critical system messages in QSYSOPR are sometimes missed because of the volume of messages sent to that message queue.

Using the History Log

Some security-related events, such as exceeding the incorrect sign-on attempts specified in the QMAXSIGN system value, cause a message to be sent to the QHST (history) log. Security messages are in the range 2200 to 22FF. They have the prefixes CPI, CPF, CPC, CPD, and CPA.

Beginning with Version 2 Release 3 of the OS/400 licensed program, some authority failure and integrity violation messages are no longer sent to the QHST (history) log. All information that was available in the QHST log can be obtained from the security audit journal. Logging information to the audit journal provides better system performance and more complete information about these security-related events than the QHST log.

These messages are no longer written to the QHST log:

- CPF2218. These events can be captured in the audit journal by specifying *AUTFAIL for the QAUDLVL system value.
- CPF2240. These events can be captured in the audit journal by specifying *AUTFAIL for the QAUDLVL system value.

Use the Display Log (DSPLOG) command to display the history log (QHST). The DSPLOG command has parameters allowing you to narrow your search to a particular range of messages and time frame. You can also specify a particular message ID, if you are looking for one type of violation.

When viewing messages from a workstation (rather than printing them), place the cursor on a message, and press the Help key. Additional information, such as the time and date of when the attempt was made, is shown.

The Print Security Violations (PRTSECVIL) tool in the QUSRTOOL library provides another method for viewing the security-related messages in the history log.

Using Journals to Monitor File Activity

If you include the *AUTFAIL value for system action auditing (the QAUDLVL system value), the system writes an audit journal entry for every unsuccessful attempt to access a resource. For critical files, you can also set up object auditing so the system writes an audit journal entry for each successful access.

The audit journal records only that the file was opened. It does not log every transaction to the file. For critical files on your system, you may want more detailed information about the specific records that were accessed and changed. The file journaling function, which is primarily used for application integrity and recovery, can also be used by the security officer or auditor to review file transactions.

A journal can include:

- Identification of the job and user and the time of access
- Before- and after-images of all file changes
- Records of when the file was opened, closed, and saved

A journal entry cannot be altered by any user, even the security officer. A complete journal can be deleted, but this is easily detected.

If you are journaling and want to print all information about a particular file, type the following:

```
DSPJRN JRN(library/journal) +
       FILE(library/file) OUTPUT(*PRINT)
```

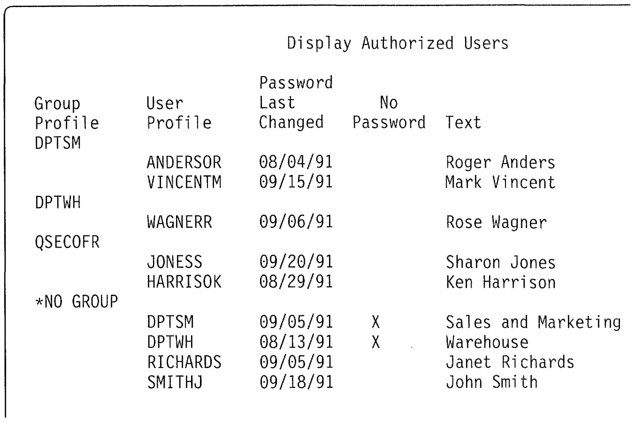
For example, if journal JRNCUST in library CUSTLIB is used to record information about file CUSTFILE (also in library CUSTLIB), the command would be:

```
DSPJRN JRN(CUSTLIB/JRNCUST) +
       FILE(CUSTLIB/CUSTFILE) OUTPUT(*PRINT)
```

If you want to find out which journals are on the system, use the Work with Journals (WRKJRN) command. If you want to find out which files are being journaled by a particular journal, use the Work with Journal Attributes (WRKJRNA) command. The *Advanced Backup and Recovery Guide* provides complete information about journaling.

Analyzing User Profiles

You can display or print a complete list of all the users on your system with the Display Authorized Users (DSPAUTUSR) command. The list can be sequenced by profile name or group profile name. Following is an example of the group profile sequence:



```
                          Display Authorized Users

                         Password
Group         User       Last         No
Profile       Profile    Changed   Password  Text
DPTSM
              ANDERSOR   08/04/91            Roger Anders
              VINCENTM   09/15/91            Mark Vincent
DPTWH
              WAGNERR    09/06/91            Rose Wagner
QSECOFR
              JONESS     09/20/91            Sharon Jones
              HARRISOK   08/29/91            Ken Harrison
*NO GROUP
              DPTSM      09/05/91     X      Sales and Marketing
              DPTWH      08/13/91     X      Warehouse
              RICHARDS   09/05/91            Janet Richards
              SMITHJ     09/18/91            John Smith
```

Printing Selected User Profiles: You can use the Display User Profile (DSPUSRPRF) command to create an output file, which you can process using Query or the DSPSECRVW (Display Security Review) tool in the QUSRTOOL library:

```
DSPUSRPRF USRPRF(*ALL) +
          TYPE(*BASIC) OUTPUT(*OUTFILE)
```

The DSPSECRVW tool can be used to provide a variety of analysis reports of your output file, such as:

- A list of all users who have both *ALLOBJ and *SPLCTL special authority.
- A list of all users sequenced by a user profile field, such as initial program or user class.

You can create query programs to produce different reports from your output file. For example:

- List all user profiles that have any special authorities by selecting records where the field UPSPAU is not equal to *NONE.
- List all users who are allowed to enter commands by selecting records where the *Limit capabilities* field (called UPLTCP in the field reference file) is equal to *NO or *PARTIAL.
- List all users who have a particular initial menu or initial program.
- List inactive users by looking at the date last sign-on field.

Examining Large User Profiles: User profiles with large numbers of authorities, appearing to be randomly spread over most of the system, can reflect a lack of security planning. Following is one method for locating large user profiles and evaluating them:

1. Use the Display Object Description (DSPOBJD) command to create an output file containing information about all the user profiles on the system:

   ```
   DSPOBJD OBJ(*ALL) OBJTYPE(*USRPRF) +
           DETAIL(*BASIC) OUTPUT(*OUTFILE)
   ```

2. Create a query program to list the name and size of each user profile, in descending sequence by size.
3. Print detailed information about the largest user profiles and evaluate the authorities and owned objects to see if they are appropriate:

   ```
   DSPUSRPRF USRPRF(user-profile-name) +
             TYPE(*OBJAUT) OUTPUT(*PRINT)
   DSPUSRPRF USRPRF(user-profile-name) +
             TYPE(*OBJOWN) OUTPUT(*PRINT)
   ```

   Some IBM-supplied user profiles are very large because of the number of objects they own. Listing and analyzing them is usually not necessary. However, you should check for programs adopting the authority of the IBM-supplied user profiles that have *ALLOBJ special authority, such as QSECOFR and QSYS. See “Analyzing Programs That Adopt Authority.”

   Appendix B provides information about all the IBM-supplied user profiles and their functions.

Analyzing Object Authorities

You can use the following method to determine who has authority to libraries on the system:

1. Use the DSPOBJD command to list all the libraries on the system:

   ```
   DSPOBJD OBJ(*ALL) OBJTYPE(*LIB) OUTPUT(*PRINT)
   ```

2. Use the Display Object Authority (DSPOBJAUT) command to list the authorities to a specific library:

   ```
   DSPOBJAUT OBJ(library-name) OBJTYPE(*LIB) +
             OUTPUT(*PRINT)
   ```

3. Use the Display Library (DSPLIB) command to list the objects in the library:

   ```
   DSPLIB LIB(library-name) OUTPUT(*PRINT)
   ```

Using these reports, you can determine what is in a library and who has access to the library. If necessary, you can use the DSPOBJAUT command to view the authority for selected objects in the library also.

Analyzing Programs That Adopt Authority

Programs that adopt the authority of a user with *ALLOBJ special authority represent a security exposure. The following method can be used to find and inspect those programs:

1. For each user with *ALLOBJ special authority, use the Display Programs That Adopt (DSPPGMADP) command to list the programs that adopt that user’s authority:

   ```
   DSPPGMADP USRPRF(user-profile-name) +
             OUTPUT(*PRINT)
   ```

   Note: The topic “Printing Selected User Profiles” shows how to list users with *ALLOBJ authority.

2. Use the DSPOBJAUT command to determine who is authorized to use each adopting program and what the public authority is to the program:

   ```
   DSPOBJAUT OBJ(library-name/program-name) +
             OBJTYPE(*PGM) OUTPUT(*PRINT)
   ```

3. Inspect the source code and program description to evaluate:

   - Whether the user of the program is prevented from excess function, such as using a command line, while running under the adopted profile.
   - Whether the program adopts the minimum authority level needed for the intended function. Applications that use program failure can be designed using the same owner profile for objects and programs. When the authority of the program owner is adopted, the user has *ALL authority to application objects. In many cases, the owner profile does not need any special authorities.

4. Verify when the program was last changed, using the DSPOBJD command:

```
DSPOBJD OBJ(library-name/program-name) +
        OBJTYPE(*PGM) DETAIL(*FULL)
```

Auditing the Security Officer's Actions

You may want to keep a record of all actions performed by users with *ALLOBJ and *SECADM special authority. You can use the action auditing value in the user profile to do this:

1. For each user with *ALLOBJ and *SECADM special authority, use the CHGUSRAUD command to set the AUDLVL to have all values that are not included in the QAUDLVL system value on your system. For example, if the QAUDLVL system value is set to *AUTFAIL, *PGMFAIL, *PRTDTA, and *SECURITY, use this command to set the AUDLVL for a security officer user profile:

   ```
   CHGUSRAUD USER((SECUSER)
             AUDLVL(*CMD *CREATE *DELETE +
                    *OBJMGT *OFCSRV *PGMADP +
                    *SAVRST *SERVICE, +
                    *SPLFDTA *SYSMGT)
   ```

   Note: Table 9-1 on page 9-6 shows all the possible values for action auditing.

2. Remove the *AUDIT special authority from user profiles with *ALLOBJ and *SECADM special authority. This prevents these users from changing the auditing characteristics of their own profiles.

   Note: You cannot remove special authorities from the QSECOFR profile. Therefore, you cannot prevent a user signed on as QSECOFR from changing the auditing characteristics of that profile. However, if a user signed on as QSECOFR uses the CHGUSRAUD command to change auditing characteristics, an AD entry type is written to the audit journal.

   It is recommended that security officers (users with *ALLOBJ or *SECADM special authority) use their own profiles for better auditing. The password for the QSECOFR profile should not be distributed.

3. Make sure the QAUDCTL system value includes *AUDLVL.

4. Use the DSPJRN command to review the entries in the audit journal using the techniques described in "Analyzing Audit Journal Entries with Query or a Program" on page 9-13.

Appendix A. Security Commands

This appendix contains the system commands related to security. You can use these commands in place of the system menus, if you prefer, by typing these commands on a command line. The commands are divided into task-oriented groups.

The *CL Reference* manual contains more detailed information about these commands. The tables in Appendix D show what object authorities are required to use these commands.

Table A-1. Commands for Working with Authority Holders

Command Name	Descriptive Name	Function
CRTAUTHLR	Create Authority Holder	Allows you to secure a file before the file exists. Authority holders are valid only for program-described database files.
DLTAUTHLR	Delete Authority Holder	Allows you to delete an authority holder. If the associated file exists, the authority holder information is copied to the file.
DSPAUTHLR	Display Authority Holder	Allows you to display all the authority holders on the system.

Table A-2. Commands for Working with Authorization Lists

Command Name	Descriptive Name	Function
ADDAUTLE	Add Authorization List Entry	Allows you to add a user to an authorization list. You specify what authority the user has to all the objects on the list.
CHGAUTLE	Change Authorization List Entry	Allows you to change users' authorities to the objects on the authorization list.
CRTAUTL	Create Authorization List	Allows you to create an authorization list.
DLTAUTL	Delete Authorization List	Allows you to delete an entire authorization list.
DSPAUTL	Display Authorization List	Allows you to display a list of users and their authorities to an authorization list.
DSPAUTLOBJ	Display Authorization List Objects	Allows you to display a list of objects secured by an authorization list.
EDTAUTL	Edit Authorization List	Allows you to add, change, and remove users and their authorities on an authorization list.
RMVAUTLE	Remove Authorization List Entry	Allows you to remove a user from an authorization list.
RTVAUTLE	Retrieve Authorization List Entry	Used in a control language (CL) program to get one or more values associated with a user on the authorization list. The command can be used with the CHGAUTLE command to give a user new authorities in addition to the existing authorities that the user already has.
WRKAUTL	Work with Authorization Lists	Allows you to work with authorization lists from a list display.

Table A-3. Commands for Working with Object Authority and Auditing

Command Name	Descriptive Name	Function
CHGOBJAUD	Change Object Auditing	Allows you to specify whether access to an object is audited.
CHGOBJOWN	Change Object Owner	Allows you to change the ownership of an object from one user to another.
DSPOBJAUT	Display Object Authority	Displays the object owner, public authority to the object, any private authorities to the object, and the name of the authorization list used to secure the object.
DSPOBJD	Display Object Description	Displays the object auditing level for the object.
EDTOBJAUT	Edit Object Authority	Allows you to add, change, or remove a user's authority for an object.
GRTOBJAUT	Grant Object Authority	Allows you to specifically give authority to named users, all users (*PUBLIC), or users of the referenced object for the objects named in this command.
RVKOBJAUT	Revoke Object Authority	Allows you to remove one or more (or all) of the authorities given specifically to a user for the named objects.
WRKOBJ	Work with Objects	Allows you to work with object authority by selecting options on a list display.
WRKOBJOWN	Work with Objects by Owner	Allows you to work with the objects owned by a user profile.

Table A-4. Commands for Working with Passwords

Command Name	Descriptive Name	Function
CHGDSTPWD	Change Dedicated Service Tools Password	Allows you to reset the DST or the QSECOFR password to the default password shipped with the system.
CHGPWD	Change Password	Allows a user to change the user's own password.
CHGUSRPRF	Change User Profile	Allows you to change the values specified in a user's profile, including the user's password.
CRTUSRPRF	Create User Profile	When you add a user to the system, you assign a password to the user.
CHKPWD	Check Password	Allows verification of a user's password. For example, if you want the user to enter the password again to run a particular application, you can use CHKPWD in your CL program to verify the password.

Table A-5. Commands for Working with User Profiles

Command Name	Descriptive Name	Function
CHGPRF	Change Profile	Allows a user to change some of the attributes of the user's own profile.
CHGUSRAUD	Change User Audit	Allows you to specify the action and object auditing for a user profile.
CHGUSRPRF	Change User Profile	Allows you to change the values specified in a user's profile such as the user's password, special authorities, initial menu, initial program, current library, and priority limit.
CRTUSRPRF	Create User Profile	Allows you to add a user to the system and to specify values such as the user's password, special authorities, initial menu, initial program, current library, and priority limit.
DLTUSRPRF	Delete User Profile	Allows you to delete a user profile from the system. This command provides an option to delete or change ownership of objects owned by the user profile.
DSPAUTUSR	Display Authorized Users	Displays or prints the following for all user profiles on the system: associated group profile (if any), whether the user profile has a password, the date the password was last changed, and the user profile text.
DSPUSRPRF	Display User Profile command	Allows you to display a user profile in several different formats.
GRTUSRAUT	Grant User Authority	Allows you to copy private authorities from one user profile to another user profile.
RTVUSRPRF	Retrieve User Profile	Used in a control language (CL) program to get and use one or more values that are stored and associated with a user profile.
WRKUSRPRF	Work with User Profiles	Allows you to work with user profiles by entering options on a list display.

Table A-6. Related User Profile Commands

Command Name	Descriptive Name	Function
DSPPGMADP	Display Programs That Adopt	Allows you to display a list of programs and SQL packages that adopt a specified user profile.
RSTAUT	Restore Authority	Allows you to restore authorities for objects held by a user profile when the user profile was saved. These authorities can only be restored after a user profile is restored with the Restore User Profile (RSTUSRPRF) command.
RSTUSRPRF	Restore User Profile	Allows you to restore a user profile and its attributes. Restoring specific authority to objects is done with the RSTAUT command after the user profile is restored. The RSTUSRPRF command also restores all authorization lists and authority holders if RSTUSRPRF(*ALL) is specified.
SAVSECDTA	Save Security Data	Saves all user profiles, authorization lists, and authority holders without using a system that is in a restricted state.
SAVSYS	Save System	Saves all user profiles, authorization lists, and authority holders on the system. A dedicated system is required to use this function.

Table A-7. Commands for Working with Auditing

Command Name	Descriptive Name	Function
CHGOBJAUD	Change Object Auditing	Allows you to specify the auditing for an object.
CHGDLOAUD	Change Document Library Object Auditing	Allows you to specify whether access is audited for a document library object.
CHGUSRAUD	Change User Audit	Allows you to specify the action and object auditing for a user profile.

Table A-8. Commands for Working with Document Library Objects

Command Name	Descriptive Name	Function
ADDDLOAUT	Add Document Library Object Authority	Allows you to give a user access to a document or folder or to secure a document or folder with an authorization list or an access code.
CHGDLOAUD	Change Document Library Object Auditing	Allows you to specify the object auditing level for a document library object.
CHGDLOAUT	Change Document Library Object Authority	Allows you to change the authority for a document or folder.
CHGDLOOWN	Change Document Library Object Owner	Transfers document or folder ownership from one user to another user.
DSPAUTLDLO	Display Authorization List Document Library Objects	Allows you to display the documents and folders that are secured by the specified authorization list.
DSPDLOAUD	Display Document Library Object Auditing	Displays the object auditing level for a document library object.
DSPDLOAUT	Display Document Library Object Authority	Allows you to display authority information for a document or a folder.
EDTDLOAUT	Edit Document Library Object Authority	Used to add, change, or remove users' authorities to a document or folder.
GRTUSRPMN	Grant User Permission	Gives permission to a user to handle documents and folders or to do office-related tasks on behalf of another user.
RMVDLOAUT	Remove Document Library Object Authority	Used to remove a user's authority to documents or folders.
RVKUSRPMN	Revoke User Permission	Takes away document authority from one user (or all users) to access documents on behalf of another user.

Note: For more information about OfficeVision/400 commands, see the *Managing OfficeVision/400** manual.

Table A-9. Commands for Working with the System Distribution Directory

Command Name	Descriptive Name	Function
ADDDIRE	Add Directory Entry	Adds new entries to the system distribution directory. The directory contains information about a user, such as the user ID and address, system name, user profile name, mailing address, and telephone number.
CHGDIRE	Change Directory Entry	Changes the data for a specific entry in the system distribution directory. The system administrator has authority to update any of the data contained in a directory entry, except the user ID, address, and the user description. Users can update their own directory entries, but they are limited to updating certain fields.
RMVDIRE	Remove Directory Entry	Removes a specific entry from the system distribution directory. When a user ID and address is removed from the directory, it is also removed from any distribution lists.
WRKDIR	Work with Directory	Provides a set of displays that allow a user to view, add, change, and remove entries in the system distribution directory.

Note: For more information about OfficeVision/400 commands, see the *Managing OfficeVision/400** manual.

Appendix B. IBM-Supplied User Profiles

This appendix contains information about the user profiles that are shipped with the system. These profiles are used as object owners for various system functions. Some system functions also run under specific IBM-supplied user profiles.

Table B-1 shows the default values that are used for all IBM-supplied user profiles and on the Create User Profile (CRTUSRPF) command. The parameters are sequenced in the order they appear on the Create User Profile display.

Table B-2 lists each IBM-supplied profile, its purpose, and any values for the profile that are different from the defaults for IBM-supplied user profiles.

Warning: Change the passwords for these IBM-supplied user profiles when you install your system: QSECOFR, QPGMR, QSYSOPR, QUSER, QSRV, and QSRVBAS. These passwords are the same for every AS/400 system and pose a security exposure until they are changed. Do not change any other values for IBM-supplied user profiles. Changing these profiles may cause system functions to fail.

Table B-1. Default Values for User Profiles

	Default Values	
User Profile Parameter	**IBM-Supplied User Profiles**	**Create User Profile Display**
Password (PASSWORD)	*NONE	*USRPRF
Set password to expired (PWDEXP)	*NO	*NO
Status (STATUS)	*ENABLED	*ENABLED
User class (USRCLS)	*USER	*USER
Assistance level (ASTLVL)	*SYSVAL	*SYSVAL
Current library (CURLIB)	*CRTDFT	*CRTDFT
Initial program (INLPGM)	*NONE	*NONE
Initial menu (INLMNU)	MAIN	MAIN
Initial menu library	*LIBL	*LIBL
Limited capabilities (LMTCPB)	*NO	*NO
Text (TEXT)	*BLANK	*BLANK
Special authority (SPCAUT)	*ALLOBJ[1] *SAVSYS[1]	*USRCLS[2]
Special environment (SPCENV)	*NONE	*NONE
Display sign-on information (DSPSGNINF)	*SYSVAL	*SYSVAL
Password expiration interval (PWDEXPITV)	*SYSVAL	*SYSVAL
Limit device sessions (LMTDEVSSN)	*SYSVAL	*SYSVAL
Keyboard buffering (KBDBUF)	*SYSVAL	*SYSVAL
Maximum storage (MAXSTG)	*NOMAX	*NOMAX
Priority limit	0	3
Job description (JOBD)	QDFTJOBD	QDFTJOBD
Job description library	*LIBL	*LIBL
Group profile (GRPPRF)	*NONE	*NONE
Owner (OWNER)	*USRPRF	*USRPRF
Group authority (GRPAUT)	*NONE	*NONE
Accounting code (ACGCDE)	*SYS	*BLANK
Document password (DOCPWD)	*NONE	*NONE
Message queue (MSGQ)	*USRPRF	*USRPRF
Delivery (DLVRY)	*NOTIFY	*NOTIFY
Severity (SEV)	00	00
Printer device (DEV)	*WRKSTN	*WRKSTN
Output queue (OUTQ)	*WRKSTN	*WRKSTN
Attention program (ATNPGM)	*SYSVAL	*SYSVAL
Sort sequence (SRTSEQ)	*SYSVAL	*SYSVAL
Language identifier (LANGID)	*SYSVAL	*SYSVAL
Country Identifier (CNTRYID)	*SYSVAL	*SYSVAL
Coded Character Set Identifier (CCSID)	*SYSVAL	*SYSVAL
User Option (USROPT)	*NONE	*NONE
Authority (AUT)	*EXCLUDE	*EXCLUDE
Action auditing (AUDLVL) [3]	*NONE	*NONE
Object auditing (OBJAUD) [3]	*NONE	*NONE

[1] When the system security level is changed from level 10 or 20 to level 30 or above, this value is removed.

[2] When a user profile is automatically created at security level 10, the *USER user class gives *ALLOBJ and *SAVSYS special authority.

[3] Action and object auditing are specified using the CHGUSRAUD command.

Table B-2. IBM-Supplied User Profiles

Profile Name	Descriptive Name	Parameters Different from Default Values
QDBSHR	Database share profile	
QDFTOWN	Default owner profile	PTYLMT: 3 ACGCDE: *BLANK
QDOC	Document profile	ACGCDE: *BLANK
QDSNX	Distributed systems node executive profile	PTYLMT: 3 CCSID: *HEX SRTSEQ: *HEX
QFNC	Finance profile	PTYLMT: 3
QGATE	VM/MVS* bridge profile	CCSID: *HEX SRTSEQ: *HEX
QLPAUTO	Licensed program automatic install profile	USRCLS: *SYSOPR INLMNU: *SIGNOFF SPCAUT: *ALLOBJ *JOBCTL *SAVSYS *SECADM INLPGM: QLPINATO Library: QSYS DLVRY: *HOLD SEV: 99
QLPINSTALL	Licensed program install profile	USRCLS: *SYSOPR DLVRY: *HOLD SPCAUT: *ALLOBJ *JOBCTL *SAVSYS *SECADM
QPGMR	Programmer profile	PASSWORD: QPGMR USRCLS: *PGMR SPCAUT: *ALLOBJ [1] *SAVSYS *JOBCTL PTYLMT: 3 ACGCDE: *BLANK
QRJE	Remote job entry profile	USRCLS: *PGMR SPCAUT: *ALLOBJ [1] *SAVSYS [1] *JOBCTL
QSECOFR	Security officer profile	PASSWORD: QSECOFR USRCLS: *SECOFR SPCAUT: *ALLOBJ *SAVSYS *JOBCTL *SECADM *SPLCTL *SERVICE *AUDIT ACGCDE: *BLANK
QSNADS	SNA distribution services profile	CCSID: *HEX SRTSEQ: *HEX
QSPL	Spool profile	
QSPLJOB	Spool job profile	
QSRV	Service profile	PASSWORD: QSRV USRCLS: *PGMR SPCAUT: *ALLOBJ [1] *SAVSYS [1] *JOBCTL *SERVICE ASTLVL: *INTERMED ATNPGM: QSCATTN Library: QSYS
QSRVBAS	Service basic profile	PASSWORD: QSRVBAS USRCLS: *PGMR SPCAUT: *ALLOBJ [1] *SAVSYS [1] *JOBCTL ASTLVL: *INTERMED ATNPGM: QSCATTN Library: QSYS
QSYS	System profile	USRCLS: *SECOFR SPCAUT: *ALLOBJ *SECADM *SAVSYS *JOBCTL *AUDIT *SPLCTL *SERVICE
QSYSOPR	System operator profile	PASSWORD: QSYSOPR USRCLS: *SYSOPR SPCAUT: *ALLOBJ [1] *SAVSYS *JOBCTL INLMNU: SYSTEM Library: *LIBL MSGQ: QSYSOPR Library: QSYS DLVRY: *BREAK SEV: 40 ACGCDE: *BLANK
QTCP	Transmission control protocol (TCP) profile	USRCLS: *QSECOFR SPCAUT: *ALLOBJ [1] *AUDIT *JOBCTL *SAVSYS *SECADM *SERVICE *SPLCTL CURLIB: QTCP MSGQ: QTCP/QTCP OUTQ: *DEV PTYLMT: 3
QTMPLPD	Transmission control protocol/Internet protocol (TCP/IP) printing support profile	PTYLMT: 3 PASSWORD: *NONE AUT: *USE
QTSTRQS	Test request profile	
QUSER	Workstation user profile	PASSWORD: QUSER PTYLMT: 3

[1] When the system security level is changed from level 10 or 20 to level 30 or above, this value is removed.

Appendix C. Commands Shipped with Public Authority *Exclude

Table C-1 identifies which commands have restricted authorization (public authority is *EXCLUDE) when your system is shipped. It shows what IBM-supplied user profiles are authorized to use these restricted commands. For more information about IBM-supplied user profiles, see the topic "IBM-Supplied User Profiles" on page 4-23.

In Table C-1, commands that are restricted to the security officer, and any user profile with *ALLOBJ authority, have an **R** in the QSECOFR profile. Commands that are specifically authorized to one or more IBM-supplied user profiles, in addition to the security officer, have an **S** under the profile names for which they are authorized).

Any commands not listed here are public, which means they can be used by all users. However, some commands require special authority, such as *SERVICE or *JOBCTL. The special authorities required for a command are listed in Appendix D.

If you choose to grant other users or the public *USE authority to these commands, update this table to indicate that commands are no longer restricted on your system. Using some commands may require the authority to certain objects on the system as well as to the commands themselves. See Appendix D for the object authorities required for commands.

Table C-1 (Page 1 of 3). Authorities of IBM-Supplied User Profiles to Restricted Commands

Command Name	QSEC-OFR	QPGMR	QSYS-OPR	QSRV	QSRV-BAS
ADDACC	R				
ADDCRSDMNK	R				
ADDDSTQ		S	S		
ADDDSTRTE		S	S		
ADDDSTSYSN		S	S		
ADDNETJOBE	R				
ADDOMSMTA		S	S	S	S
ADDOMSRTE		S	S	S	S
ADDOSIxxx [1]		S	S	S	S
ADDRPYLE		S			
ADDTCPLNK		S	S	S	S
ADDTCPPORT		S	S	S	S
ADDTCPRSI		S	S	S	S
ADDTCPRTE		S	S	S	S
ANSQST	R				
ANZPRB		S	S	S	S
ANZS34OCL	R				
APYJRNCHG		S		S	
APYPTF				S	
CFGDSTSRV		S	S		
CFGRPDS		S	S		
CFGTCP		S	S	S	S
CHGCRSDMNK	R				
CHGDSTPWD [2]	R				
CHGDSTQ		S	S		
CHGDSTRTE		S	S		
CHGJRN		S	S	S	
CHGLICINF	R				
CHGMSTK	R				
CHGNETA	R				
CHGNETJOBE	R				
CHGOMSMTA		S	S	S	S
CHGOMSRTE		S	S	S	S
CHGOSIxxx [1]		S	S	S	S
CHGPRB		S	S	S	S
CHGPTR				S	
CHGQSTDB	R				
CHGRPYLE		S			
CHGSYSLIBL	R				
CHGSYSVAL		S	S	S	
CHGS34LIBM	R				
CHGTCPA		S	S	S	S
CHGTCPLNK		S	S	S	S
CHGTCPRTE		S	S	S	S
CHKCMNTRC				S	
CHKPRDOPT		S	S	S	S
CPHDTA	R				
CPYPTF		S	S	S	S
CRTAUTHLR	R				
CRTLASREP		S			
CRTQSTDB	R				
CRTQSTLOD	R				
CVTBASSTR	R				
CVTBASUNF	R				
CVTBGUDTA	R				
CVTS36CFG	R				
CVTS36FCT	R				
CVTS36JOB	R				
CVTS36QRY	R				
CVTS38JOB	R				
DLTAPARDTA		S	S	S	S
DLTCMNTRC				S	
DLTLICPGM	R				
DLTPRB		S	S	S	S
DLTPTF		S	S	S	S
DLTQST	R				
DLTQSTDB	R				
DMPDLO		S	S	S	S
DMPJOB		S	S	S	S
DMPJOBINT		S	S	S	S
DMPOBJ		S	S	S	S
DMPSYSOBJ		S	S	S	S
DSPDSTLOG	R				
DSPOSISAP				S	S
DSPPTF		S	S	S	S

Table C-1 (Page 2 of 3). Authorities of IBM-Supplied User Profiles to Restricted Commands

Command Name	QSEC-OFR	QPGMR	QSYS-OPR	QSRV	QSRV-BAS
DSPSRVSTS		S	S	S	S
EDTQST	R				
EDTRBDAP			S		
ENCCPHK	R				
ENCFRMMSTK	R				
ENCTOMSTK	R				
ENDCMNTRC				S	
ENDCS	R				
ENDIDXMON	R				
ENDJOBABN		S	S	S	
ENDOMS		S	S	S	S
ENDOSI	R				
ENDOSIASN			S		
ENDOSINL			S		
ENDSRVJOB		S	S	S	S
ENDTCPCNN		S	S	S	S
ENDTCPLNK		S	S	S	S
GENCPHK	R				
GENCRSDMNK	R				
GENMAC	R				
GENPIN	R				
GENS36RPT	R				
GENS38RPT	R				
GRTACCAUT	R				
HLDCMNDEV		S	S	S	S
HLDDSTQ		S	S		
INZCS	R				
INZDSTQ		S	S		
INZSYS	R				
LODPTF				S	
LODQSTDB	R				
MGRS36ITM	R				
MGRS38OBJ	R				
PRTCMNTRC				S	
PRTDSKINF	R				
PRTERRLOG		S	S	S	S
PRTINTDTA		S	S	S	S
RCLSPLSTG		S	S	S	S
RCLSTG		S	S	S	S
RCLTMPSTG		S	S	S	S
RESMGRNAM	R				
RLSCMNDEV		S	S	S	S
RLSDSTQ		S	S		
RLSRMTPHS		S	S		
RMVACC	R				
RMVCRSDMNK	R				
RMVDSTQ		S	S		
RMVDSTRTE		S	S		
RMVDSTSYSN		S	S		
RMVJRNCHG		S		S	
RMVNETJOBE	R				
RMVOMSCTE		S	S	S	S
RMVOMSMTA		S	S	S	S
RMVOMSRTE		S	S	S	S
RMVOSIABSN		S	S	S	S
RMVOSIADJN		S	S	S	S

Table C-1 (Page 2 of 3). Authorities of IBM-Supplied User Profiles to Restricted Commands

Command Name	QSEC-OFR	QPGMR	QSYS-OPR	QSRV	QSRV-BAS
RMVOSIAGT			S	S	S
RMVOSIAGTR		S	S	S	S
RMVOSIAPPE		S	S	S	S
RMVOSIAPPM		S	S	S	S
RMVOSIAPPX		S	S	S	S
RMVOSIAUNN		S	S	S	S
RMVOSICLPS		S	S	S	S
RMVOSICMPS		S	S	S	S
RMVOSIDUAR		S	S	S	S
RMVOSILINE		S	S	S	S
RMVOSILINS		S	S	S	S
RMVOSIMGR			S	S	S
RMVOSIMGRR		S	S	S	S
RMVOSINSAP		S	S	S	S
RMVOSIOX25		S	S	S	S
RMVOSIQOSM		S	S	S	S
RMVOSIRTE		S	S	S	S
RMVOSISSEL		S	S	S	S
RMVOSISUBN		S	S	S	S
RMVOSITPTM		S	S	S	S
RMVPTF				S	
RMVRPYLE		S			
RMVTCPLNK		S	S	S	S
RMVTCPPORT		S	S	S	S
RMVTCPRSI		S	S	S	S
RMVTCPRTE		S	S	S	S
RSTAUT	R				
RSTCFG	R				
RSTLICPGM	R				
RSTS38AUT	R				
RSTUSRPRF	R				
RTVDSKINF	R				
RUNLPDA		S	S	S	S
SAVAPARDTA		S	S	S	S
SAVLICPGM	R				
SBMFNCJOB	R				
SETMSTK	R				
SETOSIATR			S	S	S
SNDDSTQ		S	S		
SNDPTFORD				S	S
SNDSRVRQS				S	S
STRCMNTRC				S	
STRCS	R				
STRDBG		S		S	S
STRIDXMON	R				
STROMS		S	S	S	S
STROSINL			S		
STRRGZIDX	R				
STRSAM		S		S	S
STRSRVJOB		S	S	S	S
STRSST				S	
STRS36MGR	R				
STRS38MGR	R				
STRTCPLNK		S	S	S	S

Table C-1 (Page 3 of 3). Authorities of IBM-Supplied User Profiles to Restricted Commands

Command Name	QSECOFR	QPGMR	QSYSOPR	QSRV	QSRVBAS
STRUPDIDX	R				
TRCCPIC	R				
TRCCS	R				
TRCICF	R				
TRCINT		S		S	
TRCJOB		S	S	S	S
TRCOSIASN				S	S
TRCOSIPCL				S	S
TRNPIN	R				
VFYCMN		S	S	S	S
VFYLNKLPDA		S	S	S	S
VFYMSTK	R				
VFYPIN	R				
VFYPRT		S	S	S	S
VFYTAP		S	S	S	S
WRKCNTINF				S	S
WRKDEVTBL	R				
WRKDPCQ		S	S		
WRKDSTQ		S	S		
WRKFSTAF	R				
WRKFSTPCT	R				
WRKJRN		S	S	S	
WRKLICINF	R				
WRKOMSMTA		S	S	S	S
WRKOMSMTAQ		S	S	S	S
WRKOMSRTE		S	S	S	S
WRKORDINF			S	S	
WRKPGMTBL	R				
WRKPRB		S	S	S	S
WRKSRVPVD				S	S
WRKTXTIDX	R				
WRKUSRTBL	R				

[1] The same IBM-supplied user profiles are authorized to all ADDOSI and CHGOSI commands.

[2] The CHGDSTPWD command is shipped with public authority *USE, but you must be signed on as QSECOFR to use this command. You cannot authorize other users to the command.

Appendix D. Authority Required for Objects Used by Commands

The tables in this appendix show what authority is needed for objects referenced by commands. For example, in the entry for the Change User Profile (CHGUSRPRF) command on page D-65, the table lists all the objects you need authority to, such as the user's message queue, job description, and initial program.

The tables are organized in alphabetical order according to object type. In addition, tables are included for items that are not OS/400 objects (jobs, spooled files, network attributes, and system values) and for some functions (device emulation and finance). Additional considerations (if any) for the commands are included as footnotes to the table.

Following are descriptions of the columns in the tables:

Referenced Object: The objects listed in the *Referenced Object* column are objects to which the user needs authority when using the command. See "Assumptions" for information about objects which are not listed for each command.

Authority Needed for Object: The authorities specified in the tables show the object authorities and the data authorities required for the object when using the command. Table D-1 describes the authorities that are specified in the *Authority Needed* column.

Table D-1. Description of Authority Needed Values

Authority Name	System Name	Description
Object Authorities:		
Operational	*OBJOPR	Object operational authority. Allows user to look at the object description and use the object in any way permitted by the user's data authorities to the object.
Management	*OBJMGT	Object management authority. Allows a user to specify the security for an object, move or rename an object, and add members to database files.
Existence	*OBJEXIST	Object existence authority. Allows a user to control the existence and ownership of an object.
Authorization list management	*AUTLMGT	Allows a user to add and remove users and their authorities on an authorization list.
Data Authorities:		
Read	*READ	Allows a user to display the contents of an object or run a program.
Add	*ADD	Allows a user to add entries to an object.
Update	*UPD	Allows a user to change the entries in an object.
Delete	*DLT	Allows a user to delete entries in an object.

In addition to these values, the *Authority Needed* column of the table may show system-defined subsets of these authorities. Table D-2 shows the subsets of object authorities and data authorities.

Table D-2. System-Defined Authority

Authority	*ALL	*CHANGE	*USE	*EXCLUDE
Object Authorities				
*OBJOPR	X	X	X	
*OBJMGT	X			
*OBJEXIST	X			
Data Authorities				
*READ	X	X	X	
*ADD	X	X		
*UPD	X	X		
*DLT	X	X		

For more information on these authorities and their descriptions, see "Defining How Information Can Be Accessed" on page 5-2.

Library Authority: This column shows what authority is needed for the library containing the object. If this column is blank, then *READ authority is required for the object library. If the column has an entry, such as *Add*, this authority is needed in addition to Read.

Assumptions

1. To use any command, *USE authority is required to the command. This authority is not specifically listed in the tables.
2. To access any object, you need at least *READ authority to the library containing the object. The authority requirement for the library is listed only if it is greater than *READ authority.
3. To enter any display command, you need operational authority to the IBM-supplied display file, printer output file, or panel group used by the command. These files and panel groups are shipped with public authority *USE.

General Rules

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
Change (CHG) with F4 (Prompt)	Current values	The current values are displayed if the user has authority to those values.	
Copy (CPY) where to-file is a database file	Object to be copied	Operational and read	
	CRTPF command, if CRTFILE (*YES) is specified	Operational	
	To-file, if CRTFILE (*YES) is specified[1]		Add
	To-file, if it exists and new member is added	Management, operational, and add	Add
	To-file, if file and member exist and *ADD option is specified	Operational and add	
	To-file, if file and member exist and *REPLACE option is specified	Management, operational, add, and delete	
Create (CRT)	Object to be created[2]		Add
	User profile that will own created object (either the user profile running the job or the user's group profile)	Add	
Create (CRT) if REPLACE(*YES) is specified [7]	Object to be created (and replaced)[2]	Existence, management, and read[5]	Add[6]
	User profile that will own created object (either the user profile running the job or the user's group profile)	Add	
Display (DSP) or other operation using output file (OUTPUT(*OUTFILE))	Object to be displayed	Use	
	Output file, if file does not exist[3]		Add
	Output file, if file exists and new member is added, or if *REPLACE option specified and member did not previously exist	Management, operational, and add	Add
	Output file, if file and member exist and *ADD option is specified	Operational and add	
	Output file, if file and member exist and *REPLACE option is specified	Management, operational, add, and delete	
	Format file (QAxxxxx file in QSYS), if output file does not exist	Use	
Display (DSP) using *PRINT or Work (WRK) using *PRINT	Object to be displayed	Use	
	Output queue[4]	Read	
	Printer file (QPxxxxx in QSYS)	Use	
Save (SAV) or other operation using device description	Device description	Use	
	Device file associated with device description, such as QSYSTAP for the TAP01 device description	Use	

[1] The user profile running the copy command becomes the owner of the to-file, unless the user is a member of a group profile and has OWNER(*GRPPRF). If the user's profile specifies OWNER(*GRPPRF), the group profile becomes the owner of the to-file. In that case, the user running the command must have *ADD authority to the group profile and the authority to add a member and write data to the new file. The to-file is given the same public authority, private authorities, and authorization list as the from-file.

[2] The user profile running the create command becomes the owner of the newly created object, unless the user is a member of a group profile and has OWNER(*GRPPRF). If the user's profile specifies OWNER(*GRPPRF), the group profile becomes the owner of the newly created object. Public authority to the object is controlled by the AUT parameter.

[3] The user profile running the display command becomes the owner of the newly created output file, unless the user is a member of a group profile and has OWNER(*GRPPRF). If the user's profile specifies OWNER(*GRPPRF), the group profile becomes the owner of the output file. Public authority to the output file is controlled by the CRTAUT parameter of the output file library.

[4] If the output queue is defined as OPRCTL (*YES), a user with *JOBCTL special authority does not need any authority to the output queue. A user with *SPLCTL special authority does not need any authority to the output queue.

[5] For device files, operational authority is also required.

[6] For files, read authority is also required.

[7] The REPLACE parameter is not available in the S/38 environment. REPLACE(*YES) is equivalent to using a function key from the programmer menu to delete the current object.

Commands Common for All Objects

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied user profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ALCOBJ [1,2,11]	Object	Operational	
CHGOBJD [3]	Object, if it is a file	Operational and management	
	Object, if it is not a file	Management	
CHGOBJOWN [3,4]	Object	Existence	
	Object (if file, library, subsystem description)	Existence and operational	
	Object (if authorization list)	Ownership or *ALLOBJ	
	Old user profile	Delete	
	New user profile	Add	
CHKOBJ [3]	Object	Authority specified by AUT parameter [14]	
CPROBJ	Object	Management	
CRTDUPOBJ [3,9,11]	New object		Use and add
	Object being copied, if it is an authorization list	Authorization list management	
	Based on physical file, when duplicating a logical file	Operational and management	Use
	Object being copied, all other types	Use and management	Use
	CRTSAVF command (if the object is a save file)	Use	
DCPOBJ	Object	Use	
DLCOBJ [1,11]	Object	Operational	
DMPOBJ (Q) [3]	Object	Use	
	Program and user profile	Read	
DMPSYSOBJ (Q)	Object	Use	
	Program and user profile	Read	
DSPOBJAUT [3]	Object (to see all authority information)	Management or *ALLOBJ	
	Output file	See General Rules on page D-2	See General Rules on page D-2
DSPOBJD [2]	Output file	See General Rules on page D-2	See General Rules on page D-2
EDTOBJAUT [3,5,6]	Object	Management	
	Object (if file)	Management and operational	
GRTOBJAUT [3,5,6]	Object	Management	
	Object (if file)	Management and operational	
MOVOBJ [3,12]	Object	Management	
	From-library		Change
	To-library		Add
RCLSTG (Q)			
RCLTMPSTG (Q)	Object	Management	
RNMOBJ [3,7,11]	Object	Management	Update
	Object (if authorization list)	Authorization list management	

Commands Common for All Objects

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied user profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
RSTOBJ [3,13]	Object, if it already exists in the library	Existence [8]	
	Message queues being restored to library where they already exist	Operational and existence [8]	
	User profile owning objects being created	Add [8]	
	To-library		Add [8]
	Library for saved object if VOL(*SAVVOL) is specified		Use [8]
	Save file	Use	
	Tape or diskette unit	Use	
	Tape (QSYSTAP) or diskette (QSYSDKT) file	Use [8]	
	QSYS/QPSRLDSP print file, if OUTPUT(*PRINT) specified	Use	
	Output file, if specified	See General Rules on page D-2	See General Rules on page D-2
	QSYS/QASRRSTO field reference file for output file, if an output file is specified and does not exist	Use	
RTVOBJD [2]	Object	Authority other than *EXCLUDE	
	Object (if file)	Operational	
RVKOBJAUT [3,5]	Object	Management	
	Object (if device file)	Management and operational	
SAVCHGOBJ [3]	Object	Existence [8]	
	Tape or diskette unit	Use	
	Save file, if empty	Use and add	
	Save file, if records exist in it	Use, add, and management	
	Save active message queue	Operational and add	
	Output file, if specified	See General Rules on page D-2	See General Rules on page D-2
	QSYS/QASAVOBJ field reference file for output file, if an output file is specified and does not exist	Use [8]	
	QSYS/QPSAVOBJ print file	Use [8]	
SAVOBJ [3]	Object	Existence [8]	
	Tape or diskette unit	Use	
	Save file, if empty	Use and add	
	Save file, if records exist in it	Use, add, and management	
	Save active message queue	Operational and add	
	Output file, if specified	See General Rules on page D-2	See General Rules on page D-2
	QSYS/QASAVOBJ field reference file for output file, if an output file is specified and does not exist	Use [8]	
	QSYS/QPSAVOBJ print file	Use [8]	
SAVSTG [10]			
SAVSYS [10]			
WRKOBJ	Object	Operational	
WRKOBJLCK			
WRKOBJOWN	User profile	Read	

Commands Common for All Objects

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied user profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)

1 See the OBJTYPE keyword of the ALCOBJ command for the list of object types that can be allocated and deallocated.

2 Ownership or some authority is required to the object.

3 This command cannot be used for documents or folders. Use the equivalent Document Library Object (DLO) command.

4 You must have *ALLOBJ and *SECADM special authority to change the object owner of a program that adopts authority.

5 You must be the owner or have object management authority and the authorities being granted or revoked.

6 You must be the owner or have *ALLOBJ special authority to grant object management authority.

7 This command cannot be used for user profiles, controller descriptions, device descriptions, line descriptions, documents, document libraries, folders, journals, and journal receivers.

8 If you have *SAVSYS special authority, you do not need the authority specified.

9 If the user running the CRTDUPOBJ command has OWNER(*GRPPRF) in his user profile, the owner of the new object is the group profile. To successfully copy authorities to a new object owned by the group profile, the following applies:

- The user running the command must have some private authority to the from-object.
- If the user has some private authority to the object, additional authorities can be obtained from adopted authority.
- Only authorities equal to or less than the user's authorities (including adopted authority) are copied to the new object.
- *OBJMGT authority is only copied if the user running the CRTDUPOBJ command is the object owner or has *ALLOBJ special authority. Adopted authority can be used to obtain ownership or *ALLOBJ special authority.

10 You must have *SAVSYS special authority.

11 This command cannot be used for journals and journal receivers.

12 This command cannot be used for journals and journal receivers, unless the from-library is QRCL and the to-library is the original library for the journal or journal receiver.

13 You must have *ALLOBJ special authority to specify ALWOBJDIF(*ALL).

14 To check a user's authority to an object, you must have the authority you are checking. For example, to check whether a user has existence authority for FILEB, you must have existence authority to FILEB.

Advanced Function Printing*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CRTFNTRSC	Source file	Use	
	Font resource: REPLACE(*NO)		Add
	Font resource: REPLACE(*YES)	See General Rules on page D-2	Add
CRTFORMDF	Source file	Use	
	Form definition: REPLACE(*NO)		Add
	Form definition: REPLACE(*YES)	See General Rules on page D-2	Add
CRTOVL	Source file	Use	
	Overlay: REPLACE(*NO)		Add
	Overlay: REPLACE(*YES)	See General Rules on page D-2	Add
CRTPAGDFN	Source file	Use	
	Page definition: REPLACE(*NO)		Add
	Page definition: REPLACE(*YES)	See General Rules on page D-2	Add
CRTPAGSEG	Source file	Use	
	Page segment: REPLACE(*NO)		Add
	Page segment: REPLACE(*YES)	See General Rules on page D-2	Add
DLTFNTRSC	Font resource	Existence	
DLTFORMDF	Form definition	Existence	
DLTOVL	Overlay	Existence	

Advanced Function Printing*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
DLTPAGDFN	Page definition	Existence	
DLTPAGSEG	Page segment	Existence	
DSPFNTRSCA	Font resource	Use	
WRKFNTRSC [1]	Font resource	Use	Use
WRKFORMDF [1]	Form definition	Use	Use
WRKOVL [1]	Overlay	Use	Use
WRKPAGDFN [1]	Page definition	Use	Use
WRKPAGSEG [1]	Page segment	Use	Use

[1] To use individual operations, you must have the authority required by the individual operation.

Alerts

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDALRD	Alert table	Use and add	
CHGALRD	Alert table	Use and update	
CHGALRTBL	Alert table	Change	
CRTALRTBL			Add
DLTALR	Physical file QAALERT	Use and delete	
DLTALRTBL	Alert table	Existence	
RMVALRD	Alert table	Use and delete	
WRKALR [1]	Physical file QAALERT	Use	
WRKALRD [1]	Alert table	Use	
WRKALRTBL [1]	Alert table	Read	Use

[1] To use individual operations, you must have the authority required by the individual operation.

Application Development

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
FNDSTRPDM	Source part	Read	
MRGFORMD	Form description	Read	
STRAPF [1]	Source file	Management and *CHANGE	Add
	Commands CRTPF, CRTLF, ADDPFM, ADDLFM, and RMVM	Use	
STRBGU [1]	Chart	Management and *CHANGE	
STRDFU [1]	Program (if create program option)		Add
	Program (if change or delete program option)	Existence	
	Program (if change or display data option)	Use	
	Database file (if change data option)	Operational, add, update, and delete	
	Database file (if display data option)	Use	
	Display file (if display or change data option)	Use	
	Display file (if change program option)	Use	
	Display file (if delete program option)	Existence	
STRPDM [1]			

Application Development

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
STRRLU	Source file	Read, add, update, and delete	
	Edit, add, or change a member	Operational and management	Add
	Browse a member	Operational	
	Print a prototype report	Operational	
	Remove a member	Operational and existence	
	Change type or text of member	Operational	
STRSDA	Source file	Read, add, update, and delete	
	Update and add new member	Change and management	Add
	Delete member	All	
STRSEU [1]	Source file	Read, add, update, and delete	
	Edit or change a member	Operational and management	
	Add a member	Operational and management	Add
	Browse a member	Operational	
	Print a member	Operational	
	Remove a member	Operational and existence	
	Change type or text of a member	Operational and management	
WRKGRPPDM [1,4]	Group [2]	Read	
WRKLIBPDM [1]			
WRKMBRPDM [1]	Source file	Read	
WRKOBJPDM [1]	File	Read	
WRKPARTPDM [1,4]	Part (object or source member)	Read	
WRKPRJPDM [1,4]	Project [3]	Read	

[1] To use the individual operations, you must have the authority required by the individual operation.

[2] A group corresponds to a library.

[3] A project consists of one or more groups (libraries).

[4] For more information, see the *Application Development Manager/400 User's Guide*.

AS/400 CSP/AE

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CRTCSPAPP	From-file	Use	
	To-file (physical file)	See General Rules on page D-2	See General Rules on page D-2
	DDS source file	Change and management	
	Application objects: REPLACE(*NO)		Add
	Application objects: REPLACE(*YES)	See General Rules on page D-2	Add
CRTCSPMSGF	Message file	Change	Add
	From-file	Use	
CHGCSPPGM	Program	Change and management	

AS/400 CSP/AE

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
DLTCSPMAP	Map group	Existence	
DLTCSPTBL	Table	Existence	

Authority Holder

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied user profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CRTAUTHLR (Q)	Associated object if it exists	All	
	Authority holder		Add
DLTAUTHLR	Authority holder	All	
	Associated file if a logical file	Operational, management, and existence	
DSPAUTHLR	Output file	See General Rules on page D-2	See General Rules on page D-2

Authorization List

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDAUTLE [1]	Authorization list	Authorization list management or ownership	
CHGAUTLE [1]	Authorization list	Authorization list management or ownership	
CRTAUTL			
DLTAUTL	Authorization list	Owner or *ALLOBJ	
DSPAUTL	Authorization list		
	Output file	See General Rules on page D-2	See General Rules on page D-2
DSPAUTLDLO	Authorization list	Use	
DSPAUTLOBJ [2]	Authorization list		
	Output file	See General Rules on page D-2	See General Rules on page D-2
EDTAUTL [1]	Authorization list	Authorization list management or ownership	
RMVAUTLE [1]			
RTVAUTLE [1]	Authorization list	Authorization list management or ownership	
WRKAUTL [3]			Use

[1] You must be the owner or have authorization list management authority and have the authorities being given, taken away, or retrieved.

[2] You must not be excluded (*EXCLUDE) from the list.

[3] Ownership or some authority to the object is required.

Binding Directory

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDBNDDIRE	Binding directory	Object operational and add	Use
	Module	Object operational and add	Use
	Service program	Object operational and add	Use
CRTBNDDIR	Binding directory		Add
DLTBNDDIR	Binding directory	Existence	Use
DSPBNDDIR	Binding directory	Use	Use
RMVBNDDIRE	Binding directory	Object operational and delete	Use
WRKBNDDIR [1]	Binding directory	Read	Use
WRKBNDDIRE [1]	Binding directory	Use	Use

[1] To use individual operations, you must have the authority required by the operation.

CallPath/400* Telephony

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDTELSWTE	QATYSWTE file	Operational, management, add	
CHGTELSWTE	QATYSWTE file	Operational, management, update	
DSPTELSWTE	QATYSWTE file	Operational	
ENDTELALMM	QATYSWTE file	Operational	
ENDTELCDRM	QATYSWTE file	Operational	
ENDTELCNNM	QATYSWTE file	Operational	
RMVTELSWTE	QATYSWTE file	Operational, existence, delete	
STRTELALMM	File	Change	Add
	Error file	Change	Add
	Data queue	Change	Add
STRTELCDRM	File	Change	Add
	Error file	Change	Add
	Data queue	Change	Add
	UNFDTAQ	Change	Add
STRTELCNNM	QATYSWTE file	Operational	
WRKTELSWTE [1]	QATYSWTE file	Operational	

[1] To use an individual operation, you must have the authority required by the operation.

Chart

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
DLTCHTFMT	Chart format	Existence	
DSPCHT	Chart format	Use	Use
	Database file	Use	Use
DSPGDF	Database file	Use	Use
STRBGU (Option 3) 2	Chart format	Change and existence	
WRKCHTFMT [1]	Chart format	Operational	Use

Chart

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)

[1] Ownership or some authority to the object is required.

[2] Option 3 on the BGU menu (shown when STRGBU is run) is the Change chart format option.

Class

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHGCLS	Class	Management and operational	
CRTCLS	Class		Add
DLTCLS	Object	Existence	
DSPCLS	Object	Operational	
WRKCLS [1]	Class	Operational	Use

[1] Ownership or some authority to the object is required.

Class-of-Service Description

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHGCOSD	Class-of-service description	Change	
CRTCOSD	Class-of-service description		
DLTCOSD	Class-of-service description	Existence	
DSPCOSD	Class-of-service description	Use	
WRKCOSD [1,2]	Class-of-service description	Operational	

[1] To use individual operations, you must have the authority required by the individual operation.

[2] Ownership or some authority to the object is required.

Commands

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHGCMD	Command	Management	
CHGCMDDFT	Command	Management and Use	
CRTCMD	Source file	Use	
	Command: REPLACE(*NO)		Add
	Command: REPLACE(*YES)	See General Rules on page D-2	See General Rules on page D-2
DLTCMD	Command	Existence	
DSPCMD	Command	Use	
SBMRMTCMD	Command	Operational	
	DDM file	Use	
SLTCMD [1]	Command	Operational	
WRKCMD [2]	Command	Operational	Use

[1] Ownership or some authority to the object is required.

[2] To use individual operations, you must have the authority required by the individual operation.

Common Cryptographic Architecture Services/400

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied user profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ENDCS (Q)	QTSS/QC3END program	Use	
INZCS (Q) [1]	QTSS/QC3INZ program	Use	
STRCS (Q)	QTSS/QC3STR program	Use	
	QTSS/QC3RTCMK program	Use	
	QSYS/QSYSNOMAX job queue	Use	
TRCCS (Q)	QTSS/QC3TRC program	Use	

[1] You must have *JOBCTL special authority.

Communications Side Information

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHGCSI	Communications side information object	Use and Object management	
CRTCSI	Communications side information object		Add
DLTCSI	Communications side information object	Existence	
DSPCSI	Communications side information object	Read	
WRKCSI	Communications side information objects	Use	

Configuration

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied user profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
PRTDEVADR	Controller description (CTL)	Use	
	Device description	Use	
RSTCFG (Q) [5]	Every object being restored over by a saved version	Existence [1]	
	To-library		Add [1]
	User profile owning objects being created	Add [1]	
	Tape unit	Use	
	Tape file (QSYSTAP)	Use [1]	
	Save file, if specified	Use	
	Print file (QPSRLDSP), if output(*print) is specified	Use	
	Output file, if specified	See General Rules on page D-2	See General Rules on page D-2
	QSYS/QASRRSTO field reference file, if output file is specified and it does not exist	Use	
RTVCFGSTS	Object	Operational	
RTVCFGSRC	Object	Use	
SAVCFG [2]	Save file, if empty	Use and add	
	Save file, if records exist in it	Use, add, and management	
VRYCFG [3]	Object	Use	
WRKCFGSTS [4]	Object	Operational	

Configuration

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied user profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)

[1] If you have *SAVSYS special authority, you do not need the authority specified.

[2] You must have *SAVSYS special authority.

[3] A user with *JOBCTL special authority does not need USE authority to the object.

[4] To use the individual operations, you must have the authority required by the individual operation.

[5] You must have *ALLOBJ special authority to specify ALWOBJDIF(*ALL).

Configuration List

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDCFGLE	Configuration list	Change	
CHGCFGL	Configuration list	Change	
CHGCFGLE	Configuration list	Change	
CPYCFGL	Configuration list	Use	
CRTCFGL	Configuration list		
DLTCFGL	Configuration list	Existence	
DSPCFGL	Configuration list	Use	
RMVCFGLE	Configuration list	Change	
WRKCFGL [1]	Configuration list	Operational	

[1] To use the individual operations, you must have the authority required by the individual operation.

Connection List

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDCNNLE	Connection list	Change	
CHGCNNL	Connection list	Change	
CHGCNNLE	Connection list	Change	
CRTCNNL			
DLTCNNL	Connection list	Existence	
DSPCNNL	Connection list	Use	
RMVCNNLE	Connection list	Change	
RNMCNNLE	Connection list	Change	
WRKCNNL [1]	Connection list	Operational	
WRKCNNLE [1]	Connection list	Use	

[1] To use the individual operations, you must have the authority required by the individual operation.

Controller Descriptions

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHGCTLAPPC	Controller description	Change	
	Line description (SWTLINLST)	Use	
	Connection list (CNNLSTOUT)	Use	
CHGCTLASC	Controller description	Change	
	Line description (SWTLINLST)	Use	

Controller Descriptions

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHGCTLBSC	Controller description	Change	
	Line description (SWTLINLST)	Use	
CHGCTLFNC	Controller description	Change	
	Line description (SWTLINLST)	Use	
CHGCTLHOST	Controller description	Change	
	Line description (SWTLINLST)	Use	
	Connection list (CNNLSTOUT)	Use	
CHGCTLLWS	Controller description	Change	
CHGCTLNET	Controller description	Change	
CHGCTLRTL	Controller description	Change	
	Line description (SWTLINLST)	Use	
CHGCTLRWS	Controller description	Change	
	Line description (SWTLINLST)	Use	
	Connection list (CNNLSTOUT)	Use	
CHGCTLTAP	Controller description	Change	
CHGCTLVWS	Controller	Change	
CRTCTLAPPC	Line description (LINE or SWTLINLST)	Use	
	Device description (DEV)	Use	
	Connection list (CNNLSTOUT)	Use	
	Controller description		
CRTCTLASC	Line description (LINE or SWTLINLST)	Use	
	Device description (DEV)	Use	
	Controller description		
CRTCTLBSC	Line description (LINE or SWTLINLST)	Use	
	Device description (DEV)	Use	
	Controller description		
CRTCTLFNC	Line description (LINE or SWTLINLST)	Use	
	Device description (DEV)	Use	
	Controller description		
CRTCTLHOST	Line description (LINE or SWTLINLST)	Use	
	Device description (DEV)	Use	
	Connection list (CNNLSTOUT)	Use	
	Controller description		
CRTCTLLWS	Device description (DEV)	Use	
	Controller description		
CRTCTLNET	Line description (LINE)	Use	
	Device description (DEV)	Use	
	Controller description		
CRTCTLRTL	Line description (LINE or SWTLINLST)	Use	
	Device description (DEV)	Use	
	Controller description		
CRTCTLRWS	Line description (LINE or SWTLINLST)	Use	
	Device description (DEV)	Use	
	Connection list (CNNLSTOUT)	Use	
	Controller description		
CRTCTLTAP	Device description (DEV)	Use	
	Controller description		

Controller Descriptions

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CRTCTLVWS	Device description (DEV)	Use	
	Controller description		
DLTCTLD	Controller description	Existence	
DSPCTLD	Controller description	Use	
ENDCTLRCY	Controller description	Operational	
RSMCTLRCY	Controller description	Operational	
WRKCTLD [1]	Controller description	Operational	

[1] To use the individual operations, you must have the authority required by the individual operation.

Cryptography

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied user profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDCRSDMNK (Q)	QUSRSYS/QACRKTBL *FILE	Operational and add	
	QHST message queue	Operational and add	
CHGCRSDMNK (Q)	QUSRSYS/QACRKTBL *FILE	Operational, read, update	
	QHST message queue	Operational and add	
CHGMSTK (Q)	QUSRSYS/QACRKTBL *FILE	Operational, read, update	
	QHST message queue	Operational and add	
CPHDTA (Q)			
ENCCPHK (Q)			
ENCFRMMSTK (Q)	QUSRSYS/QACRKTBL *FILE	Operational and read	
ENCTOMSTK (Q)	QUSRSYS/QACRKTBL *FILE	Operational and read	
GENCPHK (Q)	QUSRSYS/QACRKTBL *FILE	Operational and read	
GENCRSDMNK (Q)	QUSRSYS/QACRKTBL *FILE	Operational and add	
	QCRP/QPCRGENX *FILE	Operational and read	
	QHST message queue	Operational and add	
GENMAC (Q)			
GENPIN (Q)	QUSRSYS/QACRKTBL *FILE	Operational and read	
RMVCRSDMNK (Q)	QUSRSYS/QACRKTBL *FILE	Operational, read, delete	
	QHST message queue	Operational and add	
SETMSTK (Q)	QUSRSYS/QACRKTBL *FILE	Operational, read, update	
	QHST message queue	Operational and add	
TRNPIN (Q)	QUSRSYS/QACRKTBL *FILE	Operational and read	

Cryptography

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied user profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
VFYMSTK (Q)	QHST message queue	Operational and add	
VFYPIN (Q)	QUSRSYS/QACRKTBL *FILE	Operational and read	

Data Areas

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHGDTAARA [1]	Data area	Change	
CRTDTAARA [1]	Data area		Add
DLTDTAARA	Data area	Existence	
DSPDTAARA	Data area	Operational	
RTVDTAARA [2]	Data area	Operational	
WRKDTAARA [3]	Data area	Operational	Use

[1] If the create and change data area commands are run using high-level language functions, these authorities are still required although authority to the command is not.

[2] Authority is verified at run time, but not at compilation time.

[3] Ownership or some authority to the object is required.

Data Queues

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CRTDTAQ	Data queue		Add
	Target data queue for the QSNDDTAQ program	Operational and add	
	Source data queue for the QRCVDTAQ program	Use	
DLTDTAQ	Data queue	Existence	
WRKDTAQ [1,2]	Data queues	Read	Use

[1] To use individual operations, you must have the authority required by the individual operation.

[2] Ownership or some authority to the object is required.

Device Descriptions

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHGDEVAPPC	Device description	Change	
	Mode description (MODE)	Use	
CHGDEVASC	Device description	Change	
CHGDEVBSC	Device description	Change	
CHGDEVDKT	Device description	Change	
CHGDEVDSP [2]	Device description	Change	
	Printer (PRINTER)	Use	
CHGDEVFNC	Device description	Change	
CHGDEVHOST	Device description	Change	
CHGDEVINTR	Device description	Change	
CHGDEVNET	Device description	Change	

Device Descriptions

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read
CHGDEVPRT	Device description	Change	
CHGDEVRTL	Device description	Change	
CHGDEVSNPT	Device description	Change	
CHGDEVSNUF	Device description	Change	
CHGDEVTAP	Device description	Change	
CRTDEVAPPC	Controller description (CTL)	Use	
	Device description		
	Mode description (MODE)	Use	
CRTDEVASC	Controller description (CTL)	Use	
	Device description		
CRTDEVBSC	Controller description (CTL)	Use	
	Device description		
CRTDEVDKT	Device description		
CRTDEVDSP	Printer description (PRINTER)	Use	
	Controller description (CTL)	Use	
	Device description		
CRTDEVFNC	Controller description (CTL)	Use	
	Device description		
CRTDEVHOST	Controller description (CTL)	Use	
	Device description		
CRTDEVINTR	Device description		
CRTDEVNET	Controller description (CTL)	Use	
	Device description		
CRTDEVPRT	Controller description (CTL)	Use	
	Device description		
CRTDEVRTL	Controller description (CTL)	Use	
	Device description		
CRTDEVSNPT	Controller description (CTL)	Use	
	Device description		
CRTDEVSNUF	Controller description (CTL)	Use	
	Device description		
CRTDEVTAP	Controller description (CTL)	Use	
	Device description		
DLTDEVD [1]	Device description	Existence	
DSPCNNSTS	Device description	Operational	
DSPDEVD	Device description	Use	
ENDDEVRCY	Device description	Operational	
HLDCMNDEV [2]	Device description	Operational	
RLSCMNDEV	Device description	Operational	
RSMDEVRCY	Device description	Operational	
WRKDEVD [3]	Device description	Operational	

[1] To remove an associated output queue, object existence (*OBJEXIST) authority to the output queue and read authority to the QUSRSYS library are required.

[2] You must have job control (*JOBCTL) special authority and object operational authority to the device description.

[3] To use individual operations, you must have the authority required by the individual operation.

Device Emulation

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
EJTEMLOUT	Emulation device description when specified	Operational	
	Emulation device description when location specified	Operational	
ENDPRTEML	Emulation device description when specified	Operational	
	Emulation device description when location specified	Operational	
EMLPRTKEY	Emulation device description when specified	Operational	
	Emulation device description when location specified	Operational	
EML3270	Emulation device description	Operational	
	Emulation controller description	Operational	
STREML3270	Emulation device, emulation controller description, display station device, and display station controller description	Operational	
	Printer device description, user exit program, and translation tables when specified	Operational	
STRPRTEML	Emulation device description and emulation controller description	Operational	
	Printer device description, print file, message queue, job description, job queue, and translation tables when specified	Operational	
SNDEMLIGC	From-file	Operational	
TRMPRTEML	Emulation device description	Operational	

Directory and Directory Shadowing

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDDIRE [1]			
ADDDIRSHD [1]			
CHGDIRA [2]			
CHGDIRE [3]			
CHGDIRSHD [1]			
CPYFRMDIR [1]			
CPYTODIR [1]			
DSPDIR			
ENDDIRSHD [4]			
RMVDIRE [1]			
RMVDIRSHD [1]			
RNMDIRE [2]			
STRDIRSHD [4]			
WRKDIR [3,5]			
WRKDIRLOC [1,5]			
WRKDIRSHD [1,5]			

[1] You must have *SECADM special authority.

[2] You must have *SECADM or *ALLOBJ special authority.

[3] A user with *SECADM special authority can work with all directory entries. Users without *SECADM special authority can work only with their own entries.

[4] You must have *JOBCTL special authority.

[5] To use an individual operation, you must have the authority required by the operation.

Display Station Pass-Through

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ENDPASTHR			
STRPASTHR	APPC device on source system	Change	
	APPC device on target system	Change	
	Virtual controller on target system [1]	Use	
	Virtual device on target system [1,2]	Change	
	Program specified in the QRMTSIGN system value on target system, if any[1]	Use	Use
TFRPASTHR			

[1] The user profile that requires this authority is the profile that runs the pass-through batch job. For pass-through that bypasses the sign-on display, the user profile is the one specified in the remote user (RMTUSER) parameter. For pass-through that uses the normal sign-on procedure (RMTUSER(* NONE)), the user is the default user profile specified in the communications entry of the subsystem that handles the pass-through request. Generally, this is QUSER.

[2] If the pass-through is one that uses the normal sign-on procedure, the user profile specified on the sign-on display on the target system must have authority to this object.

Distribution

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied user profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDDSTQ (Q)			
ADDDSTRTE (Q)			
ADDDSTSYSN (Q)			
CFGDSTSRV (Q)			
CFGRPDS (Q)			
CHGDSTD [1]	Document [2]	Change	
CHGDSTQ (Q)			
CHGDSTRTE (Q)			
DLTDST [1]			
DSPDSTLOG (Q)			
DSPDSTSRV (Q)			
HLDDSTQ (Q)			
INZDSTQ (Q)			
QRYDST [1]	Requested file	Change	
RCVDST [1]	Requested file	Change	
	Folder	Change	
RLSDSTQ (Q)			
RMVDSTQ (Q)			
RMVDSTRTE (Q)			
RMVDSTSYSN (Q)			
SNDDST [1]	Requested file or document	Use	
SNDDSTQ (Q)			
WRKDSTQ (Q)			
WRKDPCQ (Q)			

[1] If the user is asking for distribution for another user, the user must have the authority to work on behalf of the other user.

[2] When the Distribution is filed.

Distribution List

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDDSTLE [1]			
CRTDSTL			
DLTDSTL [1]			
DSPDSTL			
RMVDSTLE [1]			
WRKDSTL [2]			

[1] You must have *SECADM special authority or own the distribution list.

[2] To use an individual operation, you must have the authority required by the operation.

Document Library Objects

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDDLOAUT	Document library object	All or owner	
CHGDLOAUD [1]			
CHGDLOAUT	Document library object	All or owner	
CHGDLOOWN	Document library object	Owner or *ALLOBJ special authority	
	Old user profile	Delete	
	New user profile	Add	
CHGDOCD [2]	Document description	Change	
CHKDLO [2]	Document library object	As required by the AUT keyword	
CHKDOC	Document	Change	
	Spelling aid dictionary	Change	
CPYDOC	From-document	Use	
	To-document, if replacing existing document	Change	
	To-folder if to-document is new	Change	
CRTDOC	In-folder	Change	
CRTFLR	In-folder	Change	
DLTDLO [3]	Document library object	All	
DLTDOCL	Document list	All [4]	
DMPDLO [15]			
DSPAUTLDLO	Authorization list	Use	
	Document library object	Use	
DSPDLOAUD	Output file, if specified	See General Rules on page D-2	See General Rules on page D-2
DSPDLOAUT	Document library object	Use or owner	
DSPDLONAM	Document library object	Use	
DSPDOC	Document	Use	
DSPFLR	Folder	Use	
EDTDLOAUT	Document library object	All or owner	
EDTDOC	Document	Change	
FILDOC [2]	Requested file	Use	
	Folder	Change	
MOVDOC	From-folder, if source document is in a folder	Change	
	From-document	All	
	To-folder	Change	

Document Library Objects

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
MRGDOC [5]	Document	Use	
	From-folder	Use	
	To-document if document is replaced	See General Rules on page D-2	See General Rules on page D-2
	To-folder if to-document is new	See General Rules on page D-2	See General Rules on page D-2
PAGDOC	Document	Change	
PRTDOC	Folder	Use	
	Document	Use	
	DLTPF, DLTF, and DLTOVR commands, if an *INDEX* instruction is specified	Use	
	CRTPF, OVRPRTF, DLTSPLF, and DLTOVR commands, if a *RUN* instruction is specified	Use	
	Save document, if SAVOUTPUT (*YES) is specified	Use	
	Save folder, if SAVOUTPUT (*YES) is specified	Use	
QRYDOCLIB [2,6]	Requested file	Use	
	Document list, if it exists	Change	
RCLDLO	Document library object		
	Internal documents or all documents and folders[16]		
RGZDLO	Document library object	Change or owner	
	DLO(*MAIL), DLO(*ANY), or DLO(*FLR) [16]		
RMVDLOAUT	Document library object	All or owner	
RNMDLO	Document library object	All	
	In-folder	Change	
RPLDOC [2]	Requested file	Read	
	Document	Change	
RSTDLO [7,8,9]	Document library object, if replacing	All [10]	
	Parent folder, if new DLO	Change [10]	
	Owning user profile, if new DLO	Add [10]	
	Tape or diskette unit	Use	
	Save file	Use	
	Output file, if specified	See General Rules on page D-2	See General Rules on page D-2
RSTS36FLR 11,12,14	S/36 folder	Use	
	To-folder	Change	
	Device file or device description	Use	
RTVDLONAM	Document library object	Use	
RTVDOC [2]	Document if checking out	Change	
	Document if not checking out	Use	
	Requested file	Change	
SAVDLO [7,13]	Document library object	All [10]	
	Tape or diskette unit	Use	
	Save file, if empty	Use and add	
	Save file, if records exist in it	Use, add, and management	
	Output file, if specified	See General Rules on page D-2	See General Rules on page D-2
WRKDOC	Folder	Use	
WRKFLR	Folder	Use	

Document Library Objects

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)

1 You must have *AUDIT special authority.

2 If the user is working on behalf of another user, the other user's authority to the object is checked.

3 The user must have *ALL authority to all the objects in the folder in order to delete the folder and all the objects in the folder.

4 If you have *ALLOBJ or *SECADM special authority, you do not need all *ALL authority to the document library list.

5 The user must have authority to the object being used as the merge source. For example, if MRGTYPE(*QRY) is specified, the user must have use authority to the query specified for the QRYDFN parameter.

6 Only objects that meet the criteria of the query and to which the user has at least *USE authority are returned in the document list or output file.

7 *SAVSYS, *ALLOBJ, or enrollment in the system distribution directory is required.

8 *SAVSYS or *ALLOBJ special authority is required to use the following parameter combination: RSTDLO DLO(*MAIL).

9 *ALLOBJ is required to specify ALWOBJDIF(*ALL).

10 If you have *SAVSYS or *ALLOBJ special authority, you do not need the authority specified.

11 You need *ALL authority to the document if replacing it. You need operational and all the data authorities to the folder if restoring new information into the folders, or you need *ALLOBJ special authority.

12 If used for a data dictionary, only the authority to the command is required.

13 *SAVSYS or *ALLOBJ special authority is required to use the following parameter combinations:

```
SAVDLO DLO(*ALL) FLR(*ANY)
SAVDLO DLO(*MAIL)
SAVDLO DLO(*CHG)
SAVDLO DLO(*SEARCH) OWNER(not *CURRENT)
```

14 You must be enrolled in the system distribution directory if the source folder is a document folder.

15 You must have *ALLOBJ special authority to dump internal document library objects.

16 You must have *ALLOBJ or *SECADM special authority.

Double-Byte Character Set

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CPYIGCTBL	Table for copy-in function	Operational	
	Table if table does not exist for copy-in function		Add
CRTIGCDCT	DBCS conversion dictionary		Add
DLTIGCDCT	DBCS conversion dictionary	Existence	
DLTIGCSRT	DBCS sort table	Existence	
DLTIGCTBL	DBCS font table	Existence	
DSPIGCDCT	DBCS conversion dictionary	Use	
EDTIGCDCT	DBCS conversion dictionary	Use and update	
	User dictionary	Add and delete	
STRCGU	DBCS sort table	Change	
	DBCS font table	Change	
STRFMA	DBCS font table, if copy-to option specified	Operational, read, add, and update	
	DBCS font table, if copy-from option specified	Operational and read	
	Font management aid work file (QGPL/QAFSVDF)	Change	

Edit Descriptions

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CRTEDTD	Edit description		
DLTEDTD	Edit description	Existence	
DSPEDTD	Edit description	Operational	

Edit Descriptions

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Reac
WRKEDTD [1]	Edit description	Operational	Use

[1] Ownership or some authority to the object is required.

Files

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read
ADDICFDEVE	ICF file	Operational and management	
ADDLFM	Logical file	Operational and management	Add
	Files referenced in DTAMBRS parameter	Operational and management	
ADDPFM	Physical file	Operational and management	Add
CHGDDMF	DDM file	Operational and management	
CHGDKTF	Diskette file	Operational and management	
	Device if device name specified in the command	Operational	
CHGDSPF	Display file	Operational and management	
	Device if device name specified	Operational	
CHGDTA	Data file	Operational, add, update, and delete	
	Program	Use	
	Display file	Use	
CHGICFDEVE	ICF file	Operational and management	
CHGICFF	ICF file	Operational and management	
CHGLF	Logical file	Operational and management	
CHGLFM	Logical file	Operational and management	
CHGPF	Physical file	Operational and management	
CHGPFM	Physical file	Operational and management	
CHGPRTF	Print file	Operational and management	
	Device if device name specified	Operational	
CHGSAVF	Save file	Operational and management	
CHGSRCPF	Source physical file	Operational and management	
CHGTAPF	Tape file	Operational and management	
	Device if device name specified	Operational	
CLRPFM	Physical file	Operational, management, and delete	
CLRSAVF	Save file	Operational and management	
COMMIT			

Files

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CPYF	From-file	Operational and read	
	To-file (device file)	Operational and read	
	To-file (physical file)	See General Rules on page D-2	See General Rules on page D-2
	Based-on file if from-file is logical file	Read	
CPYFRMDKT	From-file	Operational and read	
	To-file (device file)	Operational and read	
	To-file (physical file)	See General Rules on page D-2	See General Rules on page D-2
CPYFRMQRYF [1]	From-file	Operational and read	
	To-file (device file)	Operational and read	
	To-file (physical file)	See General Rules on page D-2	See General Rules on page D-2
CPYFRMTAP	From-file	Operational and read	
	To-file (device file)	Operational and read	
	To-file (physical file)	See General Rules on page D-2	See General Rules on page D-2
CPYSRCF	From-file	Operational and read	
	To-file (device file)	Operational and read	
	To-file (physical file)	See General Rules on page D-2	See General Rules on page D-2
CPYTODKT	To-file and from-file	Operational and read	
	Device if device name specified on the command	Operational and read	
	Based-on physical file if from-file is logical file	Read	
CPYTOTAP	To-file and from file	Operational and read	
	Device if device name is specified	Operational and read	
	Based-on physical file if from-file is logical file	Read	
CRTDDMF	DDM file: REPLACE(*NO)		Add
	DDM file: REPLACE(*YES)	See General Rules on page D-2	Add
CRTDKTF	Device if device name is specified	Operational	
	Diskette file: REPLACE(*NO)		Add
	Diskette file: REPLACE(*YES)	See General Rules on page D-2	Add
CRTDSPF	Source file	Use	
	Device if device name is specified	Operational	
	File specified in REF and REFFLD keywords	Operational	
	Display file: REPLACE(*NO)		Add
	Display file: REPLACE(*YES)	See General Rules on page D-2	Add

Files

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CRTICFF	Source file	Use	
	File specified in REF and REFFLD keywords	Operational	
	ICF file: REPLACE(*NO)		Add
	ICF file: REPLACE(*YES)	See General Rules on page D-2	Add
CRTLF	Source file	Use	
	Files specified on PFILE or JFILE keywords	Operational and management	
	Files specified on FORMAT and REFACCPTH keywords	Operational	
	Tables specified in the ALTSEQ keyword	Operational	
	Logical file		Add
	Files referenced in DTAMBRS parameter	Operational and management	
CRTPF	Source file	Use	
	Files specified in FORMAT and REFFLD keywords and tables specified in the ALTSEQ keyword	Operational	
	Physical file		Add
CRTPRTF	Source file	Use	
	Device if device name is specified	Operational	
	Files specified in the REF and REFFLD keywords	Operational	
	Print file: Replace(*NO)		Add
	Print file: Replace(*YES)	See General Rules on page D-2	Add
CRTSAVF	Save file		Add
CRTSRCPF	Source physical file		Add
CRTS36DSPF	To-file source file when TOMBR is not *NONE	All	Change
	Source file QS36SRC	Use	
	Display file: REPLACE(*NO)		Add
	Display file: REPLACE(*YES)	See General Rules on page D-2	Add
	Create Display File (CRTDSPF) command	Operational	
CRTTAPF	Tape file: REPLACE(*NO)		Add
	Tape file: REPLACE(*YES)	See General Rules on page D-2	Add
	Device if device name is specified	Operational	
DLTF	File	Operational and existence	
DSPDBR	Database file	Operational	
	Output file, if specified	See General Rules on page D-2	See General Rules on page D-2
DSPDDMF	DDM file	Operational (not read?)	
DSPDTA	Data file	Use	
	Program	Use	
	Display file	Use	
DSPFD [2]	File	Operational	
	Output file	See General Rules on page D-2	See General Rules on page D-2
	File is a physical file and TYPE(*ALL, *MBR, OR *MBRLST) is specified	One data authority (read, add, update, or delete)	

Files

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
DSPFFD	File	Operational	
	Output file	See General Rules on page D-2	See General Rules on page D-2
DSPPFM	Physical file	Use	
DSPSAVF	Save file	Use	
ENDCMTCTL	Message queue, when specified on NFYOBJ keyword	Operational and add	
	Data area, when specified on NFYOBJ keyword	Change	
	Files, when specified on NFYOBJ keyword	Operational and add	
INZPFM	Physical file, when RECORD(*DFT) is specified	Operational, management, and add	
	Physical file, when RECORD(*DLT) is specified	Operational, management, add, and delete	
OPNDBF	Database file	Operational	
OPNQRYF	Query file	Operational	
RGZPFM	File containing member	Operational, management, and all data authorities	
RMVICFDEVE	ICF file	Operational and management	
RMVM	File containing member	Existence	
RNMM	File containing member	Operational and management	Update
ROLLBACK			
RSTS36F [4]	To-file	All	See General Rules on page D-2
	From-file	Use	
	Based on physical file, if file being restored is a logical (alternative) file	Change	
	Device description for diskette or tape	Use	
RTVMBRD	File	Use	
SAVSAVFDTA	Tape or diskette device description	Use	
	Save file	Use	
SAVS36F	From-file	Use	
	To-file, when it is a physical file	All	See General Rules on page D-2
	Device file or device description	Use	
SAVS36LIBM	To-file, when it is a physical file	All	See General Rules on page D-2
	From-file	Use	
	Device file or device description	Use	
STRAPF [3]	Source file	Management and *CHANGE	Add
	Commands CRTPF, CRTLF, ADDPFM, ADDLFM, and RMVM	Use	
STRCMTCTL	Message queue, when specified on NFYOBJ keyword	Operational and add	
	Data area, when specified on NFYOBJ keyword	Change	
	Files, when specified on NFYOBJ keyword	Operational and add	

Files

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read
STRDFU [3]	Program (if create program option)		Add
	Program (if change or delete program option)	Existence	Add
	File (if change or display data option)	Operational, add, update, delete	
	File (if display data option)	Read	
UPDDTA	File	Change	
WRKDDMF [3]	DDM file	Operational, management, and existence	Add
WRKF [3,5]	Files	Operational	Use

[1] The CPYFRMQRYF command uses a FROMOPNID parameter rather than a FROMFILE parameter. A user must have sufficient authority to perform the OPNQRYF command prior to running the CPYFRMQRYF command. If CRTFILE(*YES) is specified on the CPYFRMQRYF command, the first file specified on the corresponding OPNQRYF FILE parameter is considered to be the from-file when determining the authorities for the new to-file. (See note 1 of General Rules on page D-2.)

[2] Ownership or operational authority to the file is required.

[3] To use individual operations, you must have the authority required by the individual operation.

[4] If a new file is created and an authority holder exists for the file, then the user must have all (*ALL) authority to the authority holder or be the owner of the authority holder. If there is no authority holder, the owner of the file is the user who entered the RSTS36F command and the public authority is *ALL.

[5] Ownership or some authority for the object is required.

Filter

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDALRACNE	Filter	Use and add	
ADDALRSLTE	Filter	Use and add	
ADDPRBACNE	Filter	Use and add	
ADDPRBSLTE	Filter	Use and add	
CHGALRACNE	Filter	Use and update	
CHGALRSLTE	Filter	Use and update	
CHGFTR	Filter	Management	
CHGPRBACNE	Filter	Use and update	
CHGPRBSLTE	Filter	Use and update	
CRTFTR	Filter		Add
DLTFTR	Filter	Existence	
RMVFTRACNE	Filter	Use and delete	
RMVFTRSLTE	Filter	Use and delete	
WRKFTR [1]	Filter	Any authority	
WRKFTRACNE [1]	Filter	Use	
WRKFTRSLTE [1]	Filter	Use	

[1] To use an individual operation, you must have the authority required by the operation.

Finance

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
SBMFNCJOB (Q)	Job description and message queue	Operational	
WRKDEVTBL (Q)	Device description	At least one data authority	

Finance

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
WRKPGMTBL (Q)			
WRKUSRTBL (Q)			

Graphics Symbol Set

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CRTGSS	Source file	Use	
	Graphics symbol set		Add
DLTGSS	Graphics symbol set	Existence	
WRKGSS [1]	Graphics symbol set	Operational	Use

[1] Ownership or some authority to the object is required.

Interactive Data Definitions

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDDTADFN	Data dictionary	Change	
	File	Operational and management	
CRTDTADCT	Data dictionary		Add
DLTDTADCT [3]	Data dictionary	Operational and existence	
DSPDTADCT	Data dictionary	Use	
LNKDTADFN [1]	Data dictionary	Use	
	File	Operational and management	
STRIDD			
WRKDTADCT [2]	Data dictionary	Operational	
WRKDBFIDD [2]	Data dictionary	Use [4]	
	Database file	Operational	
WRKDTADFN [1]	Data dictionary	Use and change	

[1] Authority to the data dictionary is not required to unlink a file.

[2] To use individual operations, you must have the authority required by the individual operation.

[3] Before the dictionary is deleted, all linked files are unlinked. Refer to the LNKDTADFN command for authority required to unlink a file.

[4] You need use authority to the data dictionary to create a new file. No authority to the data dictionary is needed to enter data in an existing file.

Information Search Index

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDSCHIDXE	Search index	Change	Use
	Panel group	Use	
CHGSCHIDX	Search index	Change	Use
CRTSCHIDX	Search Index		Add
DLTSCHIDX	Search index	Existence	
RMVSCHIDXE	Search index	Change	Use

Information Search Index

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
STRSCHIDX	Search index	Use	
WRKSCHIDX	Search index	Operational	Use
WRKSCHIDXE	Search index	Change	Use

Job Descriptions

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHGJOBD	Job description	Operational and management	
	User profile (USER)	Operational	
CRTJOBD	User profile (USER)	Operational	
	Job description		Add
DLTJOBD	Job description	Existence	
DSPJOBD	Job description	Operational	
WRKJOBD [1]	Job description	Operational	Use

[1] Ownership or some authority to the object is required.

Job Queues

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)	Job Queue Parameters		Special Authority
				AUTCHK	OPRCTL	
CLRJOBQ [1]	Job queue	Read, add, delete		*DTAAUT		
		Owner [2]		*OWNER		
					*YES	*JOBCTL
CRTJOBQ [1]	Job queue		Add			
DLTJOBQ	Job queue	Existence				
HLDJOBQ [1]	Job queue	Read, add, delete		*DTAAUT		
		Owner [2]		*OWNER		
					*YES	*JOBCTL
RLSJOBQ [1]	Job queue	Read, add, delete		*DTAAUT		
		Owner [2]		*OWNER		
					*YES	*JOBCTL
WRKJOBQ [1,3]	Job queue	Read, add, delete		*DTAAUT		
		Owner [2]		*OWNER		
					*YES	*JOBCTL

[1] If you have *SPLCTL special authority, you do not need any authority to the job queue.

[2] You must be the owner of the job queue.

[3] If you request to work with all job queues, your list display includes all the job queues in libraries to which you have Read authority.

Job Schedule

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDJOBSCDE	Job schedule	Change	
	Job description [1]	Use	
	Job queue [1,2]	Read	
	User profile	Use	
	Message queue [1]	Use and add	
CHGJOBSCDE [3]	Job schedule	Change	
	Job description [1]	Use	
	Job queue [1,2]	Read	
	User profile	Use	
	Message queue [1]	Use and add	
HLDJOBSCDE [3]	Job schedule	Change	
RLSJOBSCDE [3]	Job schedule	Change	
RMVJOBSCDE [3]	Job schedule	Change	
WRKJOBSCDE [4]	Job schedule	Use	

[1] Both the user profile adding the entry and the user profile under which the job will run are checked for authority to the referenced object.

[2] Authority to the job queue cannot come from adopted authority.

[3] You must have *JOBCTL special authority or have added the entry.

[4] To display the details of an entry (option 5 or print format *FULL), you must have *JOBCTL special authority or have added the entry.

Jobs

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
BCHJOB	Job description [9]	Use	
	User profile in job description [10]	Use	
	Sort sequence table [10]	Use	
	Message queue [7]	Use and add	
	Job queue [11]	Read	
	Output queue [7]	Use	
CHGACGCDE [1]			
CHGGRPA [4]	Message queue if associating a message queue with a group	Operational	
CHGJOB [1,2,3,7]	New job queue or output queue if changing a job queue or an output queue	Read	
	Sort sequence table	Use	
CHGPJ	User profile for the program start request to specify *PGMSTRRQS	Use	
	User profile and job description	Use	
DLYJOB [4]			
DSCJOB [1]			
DSPACTPJ			
DSPJOB [1]			
DSPJOBLOG [1,5]			
ENDGRPJOB			
ENDJOB [1]			
ENDJOBABN [1]			
ENDPJ [6]			
HLDJOB [1]			
RLSJOB [1]			
RRTJOB			
RTVJOBA			

Jobs

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
SBMDBJOB	Database file	Use	
	Job queue	Read	
SBMDKTJOB	Message queue	Use	
	Job queue and device description	Read	
SBMJOB [2]	Job description [9]	Use	
	Message queue [7]	Use and add	
	User profile [10]	Use	
	User profile in job description [10]	Use (at level 40)	
	Job queue [11]	Read	
	Output queue [7]	Read	
	Sort sequence table [10]	Use	
SBMNETJOB	Database file	Use	
STRPJ [6]	Subsystem description	Use	
	Program	Use	
TFRBCHJOB	Job queue	Read	
TFRGRPJOB	Initial group program	Use	
TFRJOB [8]	Job queue	Read	
	Subsystem description to which the job queue is allocated	Use	
TFRSECJOB			
WRKACTJOB			
WRKJOB [1]			
WRKSBMJOB			
WRKSBSJOB			
WRKUSRJOB			

[1] Any user can run these commands for jobs running under his own user profile. A user with job control (*JOBCTL) special authority can run these commands for any job.

[2] You must have the authority (specified in your user profile) for the scheduling priority and output priority specified.

[3] To change certain job attributes, even in the user's own job, requires job control (*JOBCTL) special authority. These attributes are RUNPTY, TIMESLICE, PURGE, DFTWAIT, and TSEPOOL.

[4] This command only affects the job in which it was specified.

[5] To display the log for a job which was run with *ALLOBJ special authority, you must also have *JOBCTL and *ALLOBJ special authority.

[6] To use this command, job control *JOBCTL special authority is required.

[7] The user profile under which the submitted job runs is checked for authority to the referenced object. The adopted authority of the user submitting or changing the job is not used.

[8] If the job being transferred is an interactive job, the following restrictions apply:

- The job queue where the job is placed must be associated with an active subsystem.
- The work station associated with the job must have a corresponding work station entry in the subsystem description associated with the new subsystem.
- The work station associated with the job must not have another job associated with it that has been suspended by means of the Sys Req (System Request) key. The suspended job must be canceled before the Transfer Job command can run.
- The job must not be a group job.

[9] Both the user submitting the job and the user profile under which the job will run are checked for authority to the referenced object.

[10] The user submitting the job is checked for authority to the referenced object.

[11] Either the user submitting the job or the user profile under which the submitted job runs must be authorized. The adopted authority of the user submitting the job is not used.

Journal Receivers

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CRTJRNRCV	Journal receiver		Add

Journal Receivers

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
DLTJRNRCV	Journal receiver	Operational, existence, and some data authority	
	Journal	Operational	
DSPJRNRCVA	Journal receiver	Operational and some data authority	
	Journal, if attached	Operational and some data authority	
WRKJRNRCV	Journal receiver	Operational	Use
	Journal receiver (Delete option)	Operational and existence	
	Journal receiver (Display Description option)	Operational and some data authority	
	Journal receiver (Change Description option)	Management	

Journals

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
APYJRNCHG	Journal	Use	
	Journal receiver	Use	
	Files whose journaled changes are being applied or removed	Change and management	
CHGJRN	Journal receiver, if specified	Use and management	
	Attached journal receiver	Use and management	
	Journal	Operational, management, and update	
CMPJRNIMG	Journal	Use	
	Journal receiver	Use	
	File	Use	
CRTJRN	Journal		Add
	Journal receiver	Use and management	
DLTJRN	Journal	Operational and existence	
DSPJRN	Journal	Use	
	Journal if FILE(*ALLFILE) is specified or the specified file has been deleted from the system	Use and existence	
	Journal receiver	Use	
	File if specified	Use	
	Output file	See General Rules on page D-2	See General Rules on page D-2
DSPJRNMNU[1]			
ENDJRNAP	Journal	Management	
	File	Operational and management	
ENDJRNPF	Journal	Management	
	File	Use and management	

Journals

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
JRNAP [2]			
JRNPF [3]			
RCVJRNE	Journal	Use	
	Journal if FILE(*ALLFILE) is specified or the specified file has been deleted from the system	Use and existence	
	Journal receiver	Use	
	File	Use	
	Exit program	Read	
RMVJRNCHG	Journal	Use	
	Journal receiver	Use	
	Files whose journaled changes are being applied or removed	Change and management	
RTVJRNE	Journal	Use	
	Journal if FILE(*ALLFILE) is specified or the specified file has been deleted from the system	Use and existence	
	Journal receiver	Use	
	File	Use	
SNDJRNE	Journal	Operational and add	
	File if specified	Operational	
STRJRNAP	Journal	Operational and management	
	File	Operational and management	
STRJRNPF	Journal	Operational and management	
	File	Operational and management	
WRKJRN [4]	Journal	Use	
	Journal receiver if receiver information is requested	Use	
	File if forward or backout recovery is requested	Change and management	
	Objects that are deleted during recovery	Existence	
WRKJRNA	Journal	Operational and some data authority	
	Journal receiver [5]	Operational and some data authority	

[1] See the WRKJRN command (this command has the same function)

[2] See the STRJRNAP command.

[3] See the STRJRNPF command.

[4] Additional authority is required for specific functions called during the operation selected. For example, to restore an object, the user needs the authority listed under the RSTOBJ command.

[5] Operational and existence authority is required for journal receivers if the option is chosen to delete receivers.

Languages

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CRTBASPGM	Source file	Use	
	Externally described device and database files referred to in source program	Operational	
	Program: REPLACE(*NO)		Add
	Program: REPLACE(*YES)	See General Rules on page D-2	Add
CRTBNDC	Source file	Use	
	Externally described device and database files referred to in source program	Operational	
	Program: REPLACE(*NO)		Add
	Program: REPLACE(*YES)	See General Rules on page D-2	Add
CRTCLD	Source file	Use	
	Locale object - REPLACE(*NO)		Add
	Locale object - REPLACE(*YES)	See General Rules on page D-2	Add
CRTCLPGM	Source file	Use	
	Externally described device and database files referred to in source program	Operational	
	Program: REPLACE(*NO)		Add
	Program: REPLACE(*YES)	See General Rules on page D-2	See General Rules on page D-2
CRTCBLPGM (COBOL/400* licensed program or S/38 environment)	Source file	Use	
	Externally described device and database files referred to in source program	Operational	
	Program: REPLACE(*NO)		Add
	Program: REPLACE(*YES)	See General Rules on page D-2	Add
CRTCMOD	Source file	Use	
	Externally described device and database files referred to in source program	Operational	
	Module: REPLACE(*NO)		Add
	Module: REPLACE(*YES)	See General Rules on page D-2	Add
CRTCPGM	Source file	Use	
	Externally described device and database files referred to in source program	Operational	
	Program: REPLACE(*NO)		Add
	Program: REPLACE(*YES)	See General Rules on page D-2	Add
CRTFTNPGM (FORTRAN/400* licensed program)	Source file	Use	
	Program: REPLACE(*NO)		Add
	Program: REPLACE(*YES)	See General Rules on page D-2	Add
CRTPASPGM	Source file	Use	
	Program: REPLACE(*NO)		Add
	Program: REPLACE(*YES)	See General Rules on page D-2	Add
CRTPLIPGM	Source file	Use	
	Externally described device and database files referred to in source program	Operational	
	Program: REPLACE(*NO)		Add
	Program: REPLACE(*YES)	See General Rules on page D-2	Add
CRTRMCPGM (RM/COBOL-85* licensed program)	Source file	Use	
	Program: REPLACE(*NO)		Add
	Program: REPLACE(*YES)	See General Rules on page D-2	Add

Languages

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read
CRTRPGPGM (RPG/400* licensed program and S/38 environment)	Source file	Use	
	Externally described device and database files referred to in source program	Operational	
	Program: REPLACE(*NO)		Add
	Program: REPLACE(*YES)	See General Rules on page D-2	Add
CRTRPTPGM (RPG/400 licensed program and S/38 environment)	Source file	Use	
	Program - REPLACE(*NO)		Add
	Program - REPLACE(*YES)	See General Rules on page D-2	Add
	Source file for generated RPG program	See General Rules for replacing and adding members on page D-2	See General Rules for replacing and adding members on page D-2
	Externally described device and database files referred to in source program	Operational	
CRTS36CBL (S/36 environment)	Source file	Use	
	Program: REPLACE(*NO)		Add
	Program: REPLACE(*YES)	See General Rules on page D-2	Add
CRTS36RPG	Source file	Use	Add
	Program: REPLACE(*NO)		Add
	Program - REPLACE(*YES)	See General Rules on page D-2	Add
CRTS36RPGR	Source file	Use	Add
	Display file: REPLACE(*NO)		Add
	Display file: REPLACE(*YES)	See General Rules on page D-2	Add
CRTS36RPT	Source file	Use	
	Report		Add
	Source file for generated RPG program	See General Rules for replacing and adding members on page D-2	See General Rules for replacing and adding members on page D-2
	Program: REPLACE(*NO)		Add
	Program - REPLACE(*YES)	See General Rules on page D-2	Add
CRTSQLC (SQL/400* licensed program) 1,2	Source file	Use	
	Data description specifications	Operational	
	Program: REPLACE(*NO)		Add
	Program: REPLACE(*YES)	See General Rules on page D-2	Add
CRTSQLCI (SQL/400 licensed program) 1,2	Source file	Use	
	Data description specifications	Operational	
	Program: REPLACE(*NO)		Add
	Program: REPLACE(*YES)	See General Rules on page D-2	Add
CRTSQLCBL (SQL/400 licensed program) 1,2	Source file	Use	
	Data description specifications	Operational	
	Program: REPLACE(*NO)		Add
	Program: REPLACE(*YES)	See General Rules on page D-2	Add

Languages

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CRTSQLFTN (SQL/400 licensed program) [1,2]	Source file	Use	
	Data description specifications	Operational	
	Program: REPLACE(*NO)		Add
	Program: REPLACE(*YES)	See General Rules on page D-2	Add
CRTSQLPLI (SQL/400 licensed program) [1,2]	Source file	Use	
	Data description specifications	Operational	
	Program: REPLACE(*NO)		Add
	Program: REPLACE(*YES)	See General Rules on page D-2	Add
CRTSQLRPG (SQL/400 licensed program) [1,2]	Source file	Use	
	Data description specifications	Operational	
	Program: REPLACE(*NO)		Add
	Program: REPLACE(*YES)	See General Rules on page D-2	Add
ENDCBLDBG (COBOL/400 licensed program or S/38 environment)	Program	Change	
ENTCBLDBG (S/38 environment)	Program	Change	
DLTCLD	Locale object	Existence and management	
RTVCLDSRC	Locale object	Use	
	To-file	See General Rules on page D-2	See General Rules on page D-2
RUNSQLSTM (SQL/400 licensed program) [1]	Source file	Use	
STRBAS	Externally described device and database files referred to in source program	Operational	
	Source file	Read, add, update, and delete	Add
STRBASPRC	Source file	Read	
STRCBLDBG	Program	Change	
STRREXPRC	Source file	Use	
STRSQL (SQL/400 licensed program) [1]	Data description specifications	Operational	
	Program		Add

[1] The *SQL/400* Reference* contains more information about security requirements for structured query language (SQL) statements.

[2] If a value is specified for RDBNAME parameter, *USE authority is needed to the CRTSQLPKG command.

Libraries

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied user profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object [1]	Library Authority [2] (if greater than Read)
ADDLIBLE	Library	Use	
CHGCURLIB	New current library	Use	
CHGLIB [8]	Library	Management	
CHGLIBL	Every library being placed in the library list	Use	
CHGSYSLIBL (Q)	Libraries in new list	Use	
CLRLIB [3]	Every object being deleted from library	Existence	

Libraries

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied user profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object [1]	Library Authority [2] (if greater than Read)
CPYLIB [4]	From-Library	Use	
	To-library, if it exists	Use and add	
	To-library, if it does not exist		Add
	CHKOBJ command	Use	
	Object being copied, if it is an authorization list	Authorization list management	
	Based-on physical file, if the object being copied is a logical file	Operational and management	
	Object being copied, all other types	Use	
CRTLIB [9]	Library		Add
DLTLIB [3]	Library	Use and existence	
	Every object in the library	Existence	
DSPLIB	Library	Use	
	Objects in the library [5]	Use	
DSPLIBD	Library	Some authority other than *EXCLUDE	
EDTLIBL	Library to add to list	Use	
RSTLIB [7]	Library, if it does not exist		Add [6]
	Message queues being restored to library where they already exist	Operational and existence [7]	
	Library saved if VOL(*SAVVOL) is specified	Use [6]	
	Every object being restored over in the library	Existence [3]	
	User profile owning objects being created	Add [6]	
	Tape or diskette unit	Use	
	Tape (QSYSTAP) or diskette (QSYSDKT) file	Use [6]	
	QSYS/QPSRLDSP print file, if OUTPUT(*PRINT) specified	Use	
	Save file	Use	
	Output file, if specified	See General Rules on page D-2	See General Rules on page D-2
	QSYS/QASRRST field reference file, if output file is specified and does not exist	Use	
RSTS36LIBM	From-file	Use	
	To-file	Change	
	To-library	Change	
	Device file or device description	Use	
RTVLIBD	Library	Some authority other than *EXCLUDE	
SAVLIB	Every object in the library	Existence [6]	
	Save file, if empty	Use and add	
	Save file, if records exist in it	Use, add, and management	
	Save active message queue	Operational and add	
	Tape or diskette unit	Use	
	Output file, if specified	See General Rules on page D-2	See General Rules on page D-2
	QSYS/QASAVOBJ field reference file, if output file is specified and does not exist	Use [6]	
	QSYS/QPSAVOBJ print file	Use [6]	

Libraries

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied user profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object [1]	Library Authority [2] (if greater than Read)
SAVS36LIBM	Save to a physical file	Operational and management	
	Either QSYSDKT for diskette or QSYSTAP for tape, and all commands need authority to the device	Operational	
	Save to a physical file if MBROPT(*ADD) is specified	Add	
	Save to a physical file if MBROPT(*REPLACE) is specified	Add and delete	
	From-library	Use	
WRKLIB	Library	Use	

[1] The authority needed for the library being acted upon is indicated in this column. For example, to add the library CUSTLIB to a library list using the ADDLIBLE command requires Use authority to the CUSTLIB library.

[2] The authority needed for the QSYS library is indicated in this column, because all libraries are in QSYS library. For example, to create a library (CRTLIB command), you must have Add authority to the QSYS library.

[3] If object existence is not found for some objects in the library, those objects are not deleted, and the library is not completely cleared and deleted. Only authorized objects are deleted.

[4] All restrictions that apply to the CRTDUPOBJ command, also apply to this command.

[5] Your list shows only those objects in the library for which you have use authority.

[6] If you have *SAVSYS special authority, you do not need the authority specified.

[7] You must have *ALLOBJ special authority to specify ALWOBJDIF(*ALL).

[8] You must have *AUDIT special authority to change the CRTOBJAUD value for a library.

[9] You must have *AUDIT special authority to specify a CRTOBJAUD value other than *SYSVAL.

Licensed Program

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied user profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHGLICINF (Q)	WRKLICINF command	Use	
DLTLICPGM [1,2] (Q)			
INZSYS (Q)			
RSTLICPGM [1,2] (Q)			
SAVLICPGM [1,2] (Q)			
WRKLICINF [3](Q)			

[1] Some licensed programs can be deleted, saved, or restored only if you are enrolled in the system distribution directory.

[2] If deleting, restoring, or saving a licensed program that contains folders, all restrictions that apply to the DLTDLO command also apply to this command.

[3] To use individual operations, you must have the authority required by the individual operation.

Line Descriptions

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHGLINASC	Line description	Change	
	Controller description (SWTCTLLST)	Use	
CHGLINBSC	Line description	Change	
	Controller description (SWTCTLLST)	Use	
CHGLINDDI	Line description	Change	
CHGLINETH	Line description	Change	
CHGLINFR	Line description	Change	

Line Descriptions

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHGLINIDLC	Connection list (CNNLSTIN)	Use	
	Network interface description (SWTNWILST)	Use	
	Controller description (CTL)	Use	
	Line description	Change	
CHGLINNET	Line description	Use	
CHGLINSDLC	Line description	Change	
CHGLINTDLC	Line description	Change	
CHGLINTRN	Line description	Change	
CHGLINX25	Line description	Change	
	Controller description (SWTCTLLST)	Use	
	Connection list (CNNLSTIN or CNNLSTOUT)	Use	
	Network interface description (SWTNWILST)	Use	
CRTLINASC	Controller description (CTL and SWTCTLLST)	Use	
	Line description		
CRTLINBSC	Controller description (SWTCTLLST and CTL)	Use	
	Line description		
CRTLINDDI	Line description		
	Network interface description (NWI)	Use	
CRTLINETH	Controller description (NETCTL)	Use	
	Line description		
	Network interface description (NWI)	Use	
CRTLINFR	Line description		
	Network interface description (NWI)	Use	
CRTLINIDLC	Connection list (CNNLSTIN)	Use	
	Network interface description (NWI or SWTNWILST)	Use	
	Controller description (CTL)	Use	
	Line description		
CRTLINNET	Network interface description (NWI)	Use	
	Controller description (CTL)	Use	
	Line description		
CRTLINSDLC	Controller description (CTL)	Use	
	Line description		
CRTLINTDLC	Controller description (WSC and CTL)	Use	
	Line description		
CRTLINTRN	Controller description (NETCTL)	Use	
	Line description		
	Network interface description (NWI)	Use	
CRTLINX25	Controller description (SWTCTLLST)	Use	
	Permanent virtual circuit (PVC) controller description (LGLCHLE)	Use	
	Line description		
	Connection list (CNNLSTIN or CNNLSTOUT)	Use	
	Network interface description (NWI or SWTNWILST)	Use	
DLTLIND	Line description	Existence	
DLTSUPQS	QAQABBPY	Read	
DSPLIND	Line description	Use	
ENDLINRCY	Line description	Operational	
RSMLINRCY	Line description	Operational	
WRKLIND [1]	Line description	Operational	

Line Descriptions

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)

[1] To use individual operations, you must have the authority required by the individual operation.

Media

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHKDKT	Diskette device description	Use	
CHKTAP	Tape device description	Use	
CLRDKT	Diskette device description	Management	
DLTDKTLBL	Diskette device description	Existence	
DMPTAP	Tape device description	Use	
DSPDKT	Diskette device description	Use	
DSPTAP	Tape device description	Use	
DUPDKT	Diskette device description	Management	
DUPTAP	Tape device description	Management	
INZDKT	Diskette device description	Management	
INZTAP	Tape device description	Management	
RNMDKT	Diskette device description	Management	

Menu and Panel Group

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHGMNU	Menu	Change	Use
CRTMNU	Source file	Use	
	Menu - REPLACE(*NO)		Add
	Menu - REPLACE(*YES)	See General Rules on page D-2	Add
CRTPNLGRP	Panel group - Replace(*NO)		Add
	Panel group - REPLACE(*YES)	See General Rules on page D-2	Add
	Source file	Use	
	Include file	Use	
CRTS36MNU	Menu - REPLACE(*NO)		Add
	Menu - REPLACE(*YES)	See General Rules on page D-2	Add
	Source file	Use	
	Message files named in source	Operational and existence	
	To-file source file when TOMBR is not *NONE	Operational, management, existence, and add	Add
	Menu display file when REPLACE(*YES) is specified	Operational and existence	
	Command text message file	Operational and existence	
	Create Message File (CRTMSGF) command	Operational	
	Add Message Description (ADDMSGD) command	Operational	
	Create Display File (CRTDSPF) command	Operational	
DLTMNU	Menu	Existence and operational	Use

Menu and Panel Group

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
DLTPNLGRP	Panel group	Existence	Use
DSPMNUA	Menu	Use	Use
GO	Menu	Use	Use
	Display file and message files with *DSPF specified	Use	
	Display file and program with *PGM specified	Use	
WRKMNU	Menu	Operational	Use
WRKPNLGRP	Panel group	Use	

Message Description

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDMSGD	Message file	Use and add	
CHGMSGD	Message file	Use and update	
DSPMSGD	Message file	Use	
RMVMSGD	Message file	Operational and delete	
RTVMSG	Message file	Use	
WRKMSGD [1]	Message file	Use	

[1] To use individual operations, you must have the authority required by the individual operation.

Message Files

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CRTMSGF	Message file		Add
DLTMSGF	Message file	Existence	
DSPMSGF	Message file	Use	
MRGMSGF	From-message file	Use	
	To-message file	Use, add, and delete	
	Replace-message file	Use, add	
WRKMSGF	Message file	Read	Use

Message Queues

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHGMSGQ	Message queue	Use and delete	
CLRMSGQ	Message queue	Operational and delete	
CRTMSGQ	Message queue		Add
DLTMSGQ	Message queue	Use, existence, and delete	
DSPLOG			
WRKMSGQ	Message queue	Read	Use

Messages

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
DSPMSG	Message queue	Use	Use
	Reply to inquiry messages	Use and add	Use
	Remove messages from message queue	Use and delete	Use
RCVMSG	Message queue	Use	
	Remove messages from queue	Use and delete	
RMVMSG	Message queue	Operational and delete	
SNDBRKMSG	Reply to inquiry messages	Operational and add	Use
SNDMSG	Message queue	Operational and add	
	Reply to inquiry messages	Operational and add	Use
SNDPGMMSG	Message queue	Operational and add	
	Message file, when sending pre-defined message	Use	
	Reply to inquiry messages	Operational and add	Use
SNDRPY	Message queue	Use and add	
	Remove messages from queue	Use, add, and delete	
SNDUSRMSG	Message queue	Operational and add	
	Message file, when sending pre-defined message	Use	
WRKMSG	Message queue	Use	Use
	Reply to inquiry messages	Use and add	Use
	Remove messages from message queue	Use and delete	Use

Migration

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
RCVMGRDTA	File	*ALL	Add
	Device	*CHANGE	
SNDMGRDTA	File	*ALL	Add
	Device	*CHANGE	

These commands do not require any object authorities:

ANZS34OCL (Q) [1]
ANZS36OCL
CHGS34LIBM (Q) [1]
CHKS36SRCA
CVTBASSTR (Q) [1]
CVTBASUNF (Q) [1]
CVTBGUDTA (Q) [1]
CVTS36CFG (Q) [1]
CVTS36FCT (Q) [1]
CVTS36JOB (Q) [1]
CVTS36QRY (Q) [1]
CVTS38JOB (Q) [1]
GENS36RPT (Q) [1]
GENS38RPT (Q) [1]
MGRS36
MGRS36APF [2]
MGRS36CBL
MGRS36DFU [2]
MGRS36DSPF
MGRS36ITM (Q) [1]
MGRS36LIB
MGRS36MNU
MGRS36MSGF
MGRS36QRY [2]
MGRS36RPG
MGRS36SEC
MGRS38OBJ (Q) [1]
MIGRATE
QMUS36
RESMGRNAM (Q) [1]
RSTS38AUT (Q) [1]
STRS36MGR (Q) [1]
STRS38MGR (Q) [1]

[1] You must have *ALLOBJ special authority.

[2] You must have *ALLOBJ special authority and have OS/400 option 4 installed.

Mode Description

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHGMODD	Mode description	Change	
CRTMODD	Mode description		
CHGSSNMAX	Device description	Operational	
DLTMODD	Mode description	Existence	
DSPMODD	Mode description	Use	
DSPMODSTS			
ENDMOD	Device description	Operational	
STRMOD	Device description	Operational	
WRKMODD [1]	Mode description	Operational	

[1] To use individual operations, you must have the authority required by the individual operation.

Module

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHGMOD	Module	Management and use	Use
	Module, if OPTIMIZE specified	Management and use	Use, add, and delete
	Module, if FRCCRT(*YES) specified	Management and use	Use, add, and delete
	Module, if RMVOBS specified	Management and use	Use
DLTMOD	Module	All	
DSPMOD	Module	Use	
WRKMOD	Module [1]	Read	Use

[1] To use individual operations, you must have the authority required by the individual operation.

Network

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied user profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDNETJOBE (Q)	User profile in the network job entry	Use	
CHGNETA [1,4]			
CHGNETJOBE (Q)	User profile in the network job entry	Use	
DLTNETF [2]			
DSPAPPNINF	Output file	See General Rules on page D-2	See General Rules on page D-2
DSPNETA			
RCVNETF [2]	To-file member does not exist, MBROPT(*ADD) specified	Use and management	Add
	To-file member does not exist, MBROPT(*REPLACE) specified	Change and management	Add
	To-file member exists, MBROPT(*ADD) specified	Use	
	To-file member exists, MBROPT(*REPLACE) specified	Change and management	
RMVNETJOBE (Q)	User profile in the network job entry	Use	
RTVNETA			
SNDNETF	Physical file or save file	Use	

Network

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied user profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
SNDNETMSG to a local user	Message queue	Operational and add	
WRKNETF [2,3]			
WRKNETJOBE [3]	QUSRSYS/QANFNJE	Use	

[1] You must have *ALLOBJ special authority.

[2] A user can run these commands on the user's own network files or on network files owned by the user's group profile. *ALLOBJ special authority is required to process network files for another user.

[3] To use an individual operation, you must have the authority required by that operation.

[4] Changing some network attributes requires *ALLOBJ and *SECADM special authorities.

Network Interface Descriptions

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHGNWIFR	Network interface description	Change	
CHGNWIISDN	Network interface description	Change	
	Line description (CHLENTRY)	Use	
CRTNWIFR	Network interface description		
	Line description (DLCI)	Use	
CRTNWIISDN	Network interface description	Use	
	Line description (CHLENTRY)	Use	
DLTNWID	Network interface description	Existence	
DSPNWID	Network interface description	Use	
WRKNWID [1]	Network interface description	Operational	

[1] To use the individual operations, you must have the authority required by the individual operation.

Node Lists

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDNODLE	Node list	Operational and add	
CRTNODL	Node list		Add
DLTNODL	Node list	Existence	
RMVNODLE	Node list	Operational, read, and delete	
WRKNODL	Node list	Use	Use
WRKNODLE	Node list	Use	

Office Services

These commands are not part of the OS/400 licensed program.

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
WRKTXTPRF	Document SYSTEM in folder	Use	

Office Services

These commands are not part of the OS/400 licensed program.

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
These commands do not require object authorities.			

ADDACC (Q)
DSPACC
DSPACCAUT
DSPIDXSTS (Q)
DSPUSRPMN
ENDIDXMON (Q)
GRTACCAUT [1,2,3] (Q)
GRTUSRPMN [1,2]
RMVACC [1] (Q)
RVKACCAUT [1]
RVKUSRPMN [1,2]
STRIDXMON (Q)
STRRGZIDX (Q)
STRUPDIDX (Q)
WRKDOCLIB [4]
WRKDOCPRTQ [5]
WRKTXTIDX (Q)

[1] You must have *ALLOBJ special authority to grant or revoke access code authority or document authority for other users.

[2] Access is restricted to documents, folders, and mail that are not personal.

[3] The access code must be defined to the system (using the Add Access Code (ADDACC) command) before you can grant access code authority. The user being granted access code authority must be enrolled in the system distribution directory.

[4] You must have *SECADM special authority.

[5] Additional authorities are required for specific functions called by the operations selected. The user also needs additional authorities for any commands called during a specific function.

Online Education

*These commands are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CVTEDU			
STREDU			

Operational Assistant

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHGBCKUP [1]	QUSRSYS/QEZBACKUPL *USRIDX	Change	
CHGCLNUP[2]	QPGMR *USRPRF	Use	
CHGPWRSCD [3]	PWRDWNSYS *CMD	Use	
CHGPWRSCDE [3]	PWRDWNSYS *CMD	Use	
DSPBCKSTS	QUSRSYS/QEZBACKUPL *USRIDX	Use	
DSPBCKUP	QUSRSYS/QEZBACKUPL *USRIDX	Use	
DSPBCKUPL	QUSRSYS/QEZBACKUPL *USRIDX	Use	
	QUSRSYS/QEZBACKUPF *USRIDX	Use	
DSPPWRSCD			
EDTBCKUPL [1]	QUSRSYS/QEZBACKUPL *USRIDX	Change	
	QUSRSYS/QEZBACKUPF *USRIDX	Change	
ENDCLNUP [4]	ENDJOB *CMD	Use	
PRTDSKINF (Q)	QUSRSYS/QAEZDISK *FILE, member QCURRENT	Use	
RTVBCKUP	QUSRSYS/QEZBACKUPL *USRIDX	Use	
RTVCLNUP			
RTVDSKINF (Q) [5]			
RTVPWRSCDE	DSPPWRSCD command	Use	

Operational Assistant

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
RUNBCKUP [1]	QUSRSYS/QEZBACKUPL *USRIDX	Use	
	QUSRSYS/QEZBACKUPF *USRIDX	Use	
	SAVLIB *CMD	Use	
	SAVCHGOBJ *CMD	Use	
	SAVDLO *CMD	Use	
	SAVSECDTA *CMD	Use	
	SAVCFG *CMD	Use	
	SAVCAL *CMD	Use	
STRCLNUP[4]	QPGMR User profile	Use	

[1] You must have *ALLOBJ or *SAVSYS special authority.

[2] You must have *ALLOBJ, *SECADM, and *JOBCTL special authorities.

[3] You must have *ALLOBJ and *SECADM special authorities.

[4] You must have *JOBCTL special authority.

[5] You must have *ALLOBJ special authority.

OSI Communications Subsystem/400

*These OSI commands are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDOSIABSN	Metatable file	Use	
ADDOSILINE	Line description	Use	
CHGOSIABSN	Metatable file	Use	
CHGOSILINE	Line description	Use	
CRTLASREP	Input file	Use	
	Metatable file	Change and management	
	Data structures file	Change and management	
	Local abstract syntax		Add

These commands do not require any object authorities:

ADDOSIADJN
ADDOSIAGT
ADDOSIAGTR
ADDOSIAPPE
ADDOSIAPPM
ADDOSIAPPX
ADDOSIAUNN
ADDOSICLPS
ADDOSICMPS
ADDOSIDUAR
ADDOSIIX25
ADDOSILINS
ADDOSIMGR
ADDOSIMGRR
ADDOSINSAP
ADDOSIOX25
ADDOSIQOSM
ADDOSIRTE
ADDOSISSEL
ADDOSISUBN
ADDOSITPTM
CHGOSIADJN
CHGOSIAPPE
CHGOSIAPPM
CHGOSIAPPX
CHGOSIAUNN
CHGOSICLPS
CHGOSICMPS
CHGOSIDUAR
CHGOSIIX25
CHGOSILCLA
CHGOSILINS
CHGOSIMGRR
CHGOSINSAP
CHGOSIOX25
CHGOSIQOSM
CHGOSIRTE
CHGOSISSEL
CHGOSISUBN
CHGOSITPTM
DSPOSISAP
ENDOSI
ENDOSIASN
ENDOSINL
RMVOSIABSN
RMVOSIADJN
RMVOSIAGT
RMVOSIAGTR
RMVOSIAPPE
RMVOSIAPPM
RMVOSIAPPX
RMVOSIAUNN
RMVOSICLPS
RMVOSICMPS
RMVOSIDUAR
RMVOSIOX25
RMVOSILINE
RMVOSILINS
RMVOSIMGR
RMVOSIMGRR
RMVOSINSAP
RMVOSIOX25
RMVOSIQOSM
RMVOSIRTE
RMVOSISSEL
RMVOSISUBN
RMVOSITPTM
SETOSIATR
STROSINL
TRCOSIASN
TRCOSIPCL

OSI Message Services (X.400)

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied profiles are authorized to the command. The security officer can grant *USE authority to others.*

These commands do not require object authorities.

ADDOMSMTA (Q)
ADDOMSRTE (Q)
CHGOMSMTA (Q)
CHGOMSRTE (Q)
DSPOMSMTA
DSPOMSRTE
ENDOMS (Q)
RMVOMSCTE (Q)
RMVOMSMTA (Q)
RMVOMSRTE (Q)
STROMS (Q)
WRKOMSMTA (Q)
WRKOMSMTAQ (Q)
WRKOMSRTE (Q)

Output Queue

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)	Output Queue Parameters		Special Authority
				AUTCHK	OPRCTL	
CHGOUTQ [1]	Data queue	Read				
	Output queue	Management, read, add, and delete		*DTAAUT		
		Owner [2]		*OWNER		
					*YES	*JOBCTL
CLROUTQ [1]	Output queue	Read, add, delete		*DTAAUT		
		Owner [2]		*OWNER		
					*YES	*JOBCTL
CRTOUTQ	Data queue	Read				
	Output queue		Add			
DLTOUTQ	Output queue	Existence				
HLDOUTQ [1]	Output queue	Read, add, delete		*DTAAUT		
		Owner [2]		*OWNER		
					*YES	*JOBCTL
RLSOUTQ [1]	Output queue	Read, add, delete		*DTAAUT		
		Owner [2]		*OWNER		
					*YES	*JOBCTL
WRKOUTQ [1,3]	Output queue	Read				
					*YES	*JOBCTL
WRKOUTQD [1,3]	Output queue	Read				
					*YES	*JOBCTL

[1] If you have *SPLCTL special authority, you do not need any authority to the output queue.

[2] You must be the owner of the output queue.

[3] If you request to work with all output queues, your list display includes all the output queues in libraries to which you have Read authority.

Packages

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CRTSQLPKG	Program	Use	
	SQL package - REPLACE(*NO)		Add
	SQL package - REPLACE(*YES)	See General Rules on page D-2	Add

Packages

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
DLTSQLPKG	Package	Existence and management	

Performance

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied profiles are authorized to the command. The security officer can grant *USE authority to others.*

These commands do not require any object authorities:

ADDPFRCOL
ANZACCGRP
ANZDBF
ANZDBFKEY
ANZPFRDTA
ANZPGM
CHGPFRCOL
CPYPFRDTA
CRTBESTMDL
CVTPFRDTA
DLTBESTMDL
DLTPFRDTA
DMPTRC
DSPACCGRP
DSPPFRDTA
DSPPFRGPH
ENDDSKCOL
ENDJOBTRC
ENDPFRCOL
ENDPFRMON
ENDSAM
ENDSAMCOL
PRTACTRPT
PRTCPTRPT
PRTDSKRPT
PRTJOBRPT
PRTJOBTRC
PRTLCKRPT
PRTPOLRPT
PRTRSCRPT
PRTSAMDTA
PRTSYSRPT
PRTTNSRPT
STRBEST
STRDSKCOL
STRJOBTRC
STRPRFG
STRPFRCOL
STRPFRMON
STRPFRT
STRSAM (Q)
STRSAMCOL
WRKPFRCOL
WRKSYSACT

Print Descriptor Group

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHGPDGPRF	User profile	Management	
CRTPDG	Print descriptor group		Add
DLTPDG	Print descriptor group	Use	Use
DSPPDGPRF	User profile	Management	
RTVPDGPRF	User profile	Read	

Problem

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDPRBACNE	Filter	Use and add	
ADDPRBSLTE	Filter	Use and add	
ANZPRB (Q)			
CHGPRB (Q)			
CHGPRBACNE	Filter	Use and update	
CHGPRBSLTE	Filter	Use and update	
DLTPRB (Q)			
DSPPRB	Output file	See General Rules on page D-2	See General Rules on page D-2
VFYCMN (Q)	Line description [1]	Use	
	Controller description [1]	Use	
	Network ID [1]	Use	
VFYTAP (Q)	Device description	Use	
VFYPRT (Q)	Device description	Use	
WRKPRB (Q)	Line, controller, and device based on problem analysis action	Use and add	

Problem

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)

[1] You need use authority to the communications object you are verifying.

Programs

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
The object authorities required for the CRTxxxPGM commands are listed in the Languages table on page D-33			
ADDBKP [1]			
ADDPGM [1,2]	Program	Change	Add
ADDTRC [1]			
CALL [1]	Program	Operational and one data authority	
CHGDBG	Debug operation	Use, add, and delete	
	Program	Change	
CHGHLLPTR [1]			
CHGPGM	Program	Management and use	Use
	Program, if recreate option specified	Management and use	Use, update, and delete
	Program, if USRPRF or USEADPAUT parameter is being changed	Owner [7]	Use, add, and delete
CHGPGMVAR [1]			
CHGPTR [1]			
CHGSRVPGM	Service program	Change	Use
	Service program, if RMVOBS specified	Change	Use
	Service program, if OPTIMIZE or FRCCRT(*YES) specified	Change	Use, add, and delete
	Service program, if USRPRF or USEADPAUT specified	Owner [7]	Use, add, and delete
CLRTRCDTA [1]			
CRTPGM	Program, Replace(*NO)	See General Rules on page D-2	Add
	Program, Replace(*YES)	See General Rules on page D-2	Add
	Service program	Use	
	Module	Use	
	Binding directory	Use	
CRTSRVPGM	Service program, Replace(*NO)	See General Rules on page D-2	Add
	Service program, Replace(*YES)	See General Rules on page D-2	Add
	Module	Use	
	Service program	Use	
	Export source file	Use	
	Binding directory	Use	
CVTCLSRC	From-file	Use	
	To-file	Operational, management, use, add, and delete	Add
DLTDFUPGM	Program	Existence	
	Display file	Existence	
DLTPGM	Program	Existence	

Programs

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
DLTSRVPGM	Service program	All	
DMPCLPGM	CL program	Read	None[3]
DSPBKP[1]			
DSPDBG[1]			
DSPMODSRC	Source file	Use	
	Any include files	Use	
DSPPGM	Program	Read	
	Program, if DETAIL(*MODULE) specified	Use	
DSPPGMREF	Program	Operational	
	Output file	See General Rules on page D-2	See General Rules on page D-2
DSPPGMVAR[1]			
DSPSRVPGM	Service program	Read	
	Service program, if DETAIL(*MODULE) specified	Use	
DSPTRC[1]			
DSPTRCDTA[1]			
ENDCBLDBG (COBOL/400 licensed program or S/38 environment)	Program	Change	
ENTCBLDBG (S/38 environment)	Program	Change	
ENDDBG[1]			
ENDRQS[1]			
EXTPGMINF	Source file and database files	Operational	
	Program information		Add
PRTCMDUSG	Program	Use	
RMVBKP[1]			
RMVPGM[1]			
RMVTRC[1]			
RSMBKP[1]			
RTVCLSRC	Program	Use and management	
	Database source file	Operational, management, add, and delete	
SETATNPGM	Attention-key-handling program	Operational or one or more data authorities	
SETPGMINF	Database files	Operational	
	Source file	Use	
	Program information		Add
	Program for which the control information is being set	Read and update	
STRCBLDBG	Program	Change	
STRDBG[1,2]	Program	Change	
	Source file[4]	Use	
	Any include files[4]		
STRSQL (SQL/400 licensed program)[5]	Data description specifications	Operational	
	Program		Add
TFRCTL	Program	Use or one data authority	
	Some language functions when using high-level languages	Read	

Programs

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
WRKPGM	Program	Operational	Use
WRKSRVPGM [6]	Service program	Read	Use

[1] When a program is in a debug operation, no further authority is needed for debug commands.

[2] If you have *SERVICE special authority, you need only use authority to the program.

[3] The DMPCLPGM command is requested from within a CL program that is already running. Because authority to the library containing the program is checked at the time the program is called, authority to the library is not checked again when the DMPCLPGM command is run.

[4] Applies only to ILE programs.

[5] The *SQL/400* Reference* contains more information about security requirements for SQL statements.

[6] To use individual operations, you need the authority required by the individual operation.

[7] You must own the program or have *ALLOBJ and *SECADM special authorities.

Queries

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ANZQRY	Query definition	Use	
CRTQMFORM	Query management form: REPLACE(*NO)		Add
	Query management form: REPLACE(*YES)	All	Add
	Source file	Use	
CRTQMQRY	Query management query: REPLACE(*NO)		Add
	Query management query: REPLACE(*YES)	All	Add
	Source file	Use	
	OVRDBF command	Use	
DLTQMFORM	Query management form	All	Update
DLTQMQRY	Query management query	All	Update
DLTQRY	Query definition	Existence	
RTVQMFORM	Query manager form	Use	
	Target source file	All	Add
	ADDPFM, CHGPFM, CLRPFM, CPYSRCF, CRTPRTF, CRTSRCPF, DLTF, DLTOVR, OVRDBF, RMVM commands	Use	
RTVQMQRY	Query manager query	Use	
	Target source file	All	Add
	ADDPFM, CHGPFM, CLRPFM, CPYSRCF, CRTPRTF, CRTSRCPF, DLTF, DLTOVR, OVRDBF, RMVM commands	Use	
RUNQRY	Query definition	Use	
	Input files	Use	
STRQMQRY [1]	Query management query	Use	
	Query management form, if specified	Use	
	Query definition, if specified	Use	
	Output file	See General Rules on page D-2	See General Rules on page D-2
	ADDPFM, CHGOBJD, CHGPFM, CLRPFM, CPYSRCF, CRTPRTF, CRTSRCPF, DLTF, DLTOVR, GRTOBJAUT OVRDBF, OVRPRTF RMVM commands (if OUTPUT(*OUTFILE) is specified)	Use	
STRQMPRC [1]	Source file containing query manager procedure	Use	
	Source file containing command source file, if specified	Use	
	OVRPRTF command, if statements result in printed report or query object.	Use	
STRQRY			
WRKQMFORM [2]	Query management form	Operational	Use
WRKQMQRY [2]	Query management query	Operational	Use

Queries

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
WRKQRY [3]			

[1] To run a query, you must have the authority required by the statements in the query. For example, to insert a row in a table requires operational and add authority to the table.

[2] Ownership or some authority to the object is required.

[3] To use individual operations, you must have the authority required by the individual operation.

Question and Answer

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ANSQST (Q)	Database file QAQAxxBQPY [1]	Read	
ASKQST	Database file QAQAxxBBPY [1] or QAQAxxBQPY [1]	Read	
CHGQSTDB (Q)	Database file QAQAxxBQPY [1]	Read	
CRTQSTDB [2] (Q)	Database files		Add
CRTQSTLOD (Q)	Database file QAQAxxBQPY [1]	Read	
DLTQST (Q)	Database file QAQAxxBQPY [1]	Read	
DLTQSTDB (Q)	Database file QAQAxxBQPY [1]	Read	
EDTQST (Q)	Database file QAQAxxBQPY [1]	Read	
LODQSTDB [2] (Q)	Database file QAQAxxBQPY [1,3]	Read	Add
STRQST [4]	Database file QAQAxxBBPY [1] or QAQAxxBQPY [1]	Read	
WRKQST	Database file QAQAxxBBPY [1] QAQAxxBQPY [1]	Read	
WRKPRDINF			

[1] The "xx" portion of the file name is the index of the Question and Answer database being operated on by the command. The index is a two-digit number in the range 00 to 99. To obtain the index for a particular Question and Answer database, use the WRKCNTINF command.

[2] The user profile running the command becomes the owner of newly created files, unless the OWNER parameter of the user's profile is *GRPPRF. Public authority for new files, except QAQAxxBBPY, is set to *EXCLUDE. Public authority for QAQAxxBBPY is set to *READ.

[3] Authority to the file is required only if loading a previously existing Question and Answer database.

[4] The command displays the Question and Answer menu. To use individual options, you must have the authority required by those options.

Reader

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ENDRDR [1]			
HLDRDR [1]			
RLSRDR [1]			
STRDBRDR	Message queue	Operational and add	
	Database file and job queue	Read	
STRDKTRDR	Message queue	Operational and add	
	Job queue and device description	Read	

[1] You must be the user who started the reader, or you must have all object (*ALLOBJ) or job control (*JOBCTL) special authority.

Relational Database

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDRDBDIRE			
CHGRDBDIRE			
DSPRDBDIRE	CRTPF command if OUTPUT(*OUTFILE) is specified and it does not exist	Operational	
	Output file, if specified	See General Rules on page D-2	See General Rules on page D-2
ENDRDBRQS			
RMVRDBDIRE			
WRKRDBDIRE			

Resource

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
DSPHDWRSC			
DSPLCLHDW			
DSPSFWRSC [1]			
EDTDEVRSC			
WRKHDWRSC [2]			

[1] The command uses adopted authority to access objects. Authority to use the command is sufficient to access all objects used by the command.

[2] If you use the option to create a configuration object, you must have authority to use the appropriate CRT command.

RJE (Remote Job Entry)

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDFCTE	Forms control table	Management, read, add, and delete	
	Device file [1,2]	Read	
	Physical file [1,2] (RJE generates members)	Management, read, and add	Add
	Physical file [1,2] (member specified)	Add	
	Program [1,2]	Read	
	Message queue [1,2]	Read and add	
	QUSER user profile	Read	
ADDRJECMNE	Session description	Read, add, and delete	
	BSC/CMN file [1,2]	Read	
	Device description [2]	Read	
	QUSER user profile	Read	
ADDRJERDRE	Session description	Read, add, and delete	
	Job queue [2]	Read	
	Message queue [2]	Read and add	

RJE (Remote Job Entry)

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDRJEWTRE	Session description	Read, add, and delete	
	Device file [1,2]	Read	
	Physical file [1,2] (RJE generates members)	Management, read, and add	Add
	Physical file [1,2] (member specified)	Add	
	Program [1,2]	Read	
	Message queue [1,2]	Read and add	
	QUSER user profile	Read	
CHGFCT	Forms control table	Operational and management	
CHGFCTE	Forms control table	Read, add, and delete	
	Device file [1,2]	Read	
	Physical file [1,2] (RJE generates members)	Management, read, and add	Add
	Physical file [1,2] (member specified)	Read and add	
	Program [1,2]	Read	
	Message queue [1,2]	Read and add	
	QUSER user profile	Read	
CHGRJECMNE	Session description	Read, add, and delete	
	BSC/CMN file [1,2]	Read	
	Device description [2]	Read	
	QUSER user profile	Read	
CHGRJERDRE	Session description	Read, add, and delete	
	Job queue [2]	Read	
	Message queue [2]	Read and add	
CHGRJEWTRE	Session description	Read, add, and delete	
	Device File [1,2]	Read	
	Physical file [1,2] (RJE generates members)	Management, read, and add	Add
	Physical file [1,2] (member specified)	Read and add	
	Program [1,2]	Read	
	Message queue [1,2]	Read and add	
	QUSER user profile	Read	
CHGSSND	Session description	Read, update and management	
	Job queue [1,2]	Read	
	Message queue [1,2]	Read and add	
	Forms control table [1,2]	Read	
	QUSER user profile	Read	
CNLRJERDR	Session description	Read	
	Message queue	Read and add	
CNLRJEWTR	Session description	Read	
	Message queue	Read and add	
CRTFCT	Forms control table		Add

RJE (Remote Job Entry)

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CRTRJEBSCF	BSC file		Add
	Source physical file (DDS)	Read	
	Device description	Read	
CRTRJECFG	Session description		Add and update
	Job queue		Add
	Job description		Add
	Subsystem description		Add
	Message queue		Add
	CMN file		Add
	BSC file		Add
	Printer file		Add
	Physical file		Add
	User profile QUSER [3]	Read	
	Output queue	Read	
	Forms control table	Read	
	Device description		Add
	Controller description		Add
	Line description		Add
	QRJE/QRJESRC	Read, update, add, and delete	Add and update
CRTRJECMNF	Communication file		Add
	Source physical file (DDS)	Read	
	Device description	Read	
CRTSSND	Session description		Add and update
	Job queue [1,2]	Read	
	Message queue [1,2]	Read and add	
	Forms control table [1,2]	Read	
	QUSER user profile	Read	
CVTRJEDTA	Forms control table	Read	
	Input file	Read	
	Output file (RJE generates member)	Management, read, and add	Add
	Output file (member specified)	Read and add	
DLTFCT	Forms control table	Existence	
DLTRJECFG	Session description	Existence	
	Job queue	Existence	
	BSC/CMN file	Existence	
	Physical file	Existence	
	Printer file	Existence	
	Message queue	Existence	
	Job description	Existence	
	Subsystem description	Existence	
	Device description [4]	Existence	
	Controller description [4]	Existence	
	Line description [4]	Existence	
DLTSSND	Session description	Existence	
DSPRJECFG	Session description	Read	
ENDRJESSN [5]	Session description	Read	

RJE (Remote Job Entry)

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
RMVFCTE	Forms control table	Read, update, and delete	
RMVRJECMNE	Session description	Read, update, and delete	
RMVRJERDRE	Session description	Read, update, and delete	
RMVRJEWTRE	Session description	Read, update, and delete	
SBMRJEJOB	Session description	Read	
	Input file [6]	Read	
	Message queue	Read and add	
	Job-related objects [7]		
STRRJECSL	Session description	Read	
	Message queue	Read	
STRRJERDR	Session description	Read	
STRRJESSN [5]	Session description	Read	Add
	Program	Read	
	User profile QUSER	Read	
	Job-related objects [7]		
STRRJEWTR	Session description	Read	
	Program [1]	Read	
	Device file [1]	Read	
	Physical file [1] (RJE generates members)	Management, read, and add	Add
	Physical file [1] (member specified)	Read and add	
	Message queue [1]	Read and add	
	QUSER user profile	Read	
WRKFCT [8]	Forms control table	Read	
WRKRJESSN [8]	Session description	Read	
WRKSSND [8]	Session description	Read	

[1] User profile QUSER requires authority to this object.

[2] If the object is not found or the required authority is not held, an information message is sent and the function of the command is still performed.

[3] This authority is required to create job description QRJESSN.

[4] This authority is only required when DLTCMN(*YES) is specified.

[5] You must have *JOBCTL special authority.

[6] Input files include those imbedded using the .. READFILE control statement.

[7] Refer to authority required for the SBMJOB command on page D-30.

[8] To use an individual operation, you must have the authority required by the operation.

Service

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied profiles are authorized to the command. The security officer can grant Use authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
APYPTF (Q)	Product information		Object management
CHKCMNTRC [3] (Q)			
CHKPRDOPT (Q)	All objects in product option [4]		

Service

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied profiles are authorized to the command. The security officer can grant Use authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CPYPTF [2] (Q)	From file	Use	Use
	To-file (physical file) [8]	See General Rules on page D-2	See General Rules on page D-2
	Tape or diskette unit	Use	
	Licensed program		Use
	Commands: CHKTAP, CPYFRMTAP, CPYTOTAP, CRTLIB, CRTSAVF, CRTTAPF, and OVRTAPF	Use	
	QSRV library	Use	
DLTAPARDTA (Q)	Commands: CLROUTQ, DLTLIB	Use	
DLTCMNTRC [3] (Q)	NWID (network ID) or line description	Use	
DLTPTF (Q)	Cover letter file [4]		
	PTF save file [4]		
DMPJOB (Q)			
DMPJOBINT (Q)			
DSPPTF (Q)	Output file	See General Rules on page D-2	See General Rules on page D-2
DSPSRVSTS (Q)			
ENDCMNTRC [3] (Q)	NWID or line description	Use	
ENDCPYSCN	Device description	Use	
ENDSRVJOB (Q)			
LODPTF [2] (Q)	Tape or diskette	Use	
PRTCMNTRC [3] (Q)	NWID (network ID) or line description	Use	
	Output file	See General Rules on page D-2	See General Rules on page D-2
PRTERRLOG (Q)			
PRTINTDTA (Q)			
RMVPTF (Q)	Object	Operational and management	
	Product information		Management
RUNLPDA (Q)	Line description	Read	
SAVAPARDTA [6] (Q)	Commands: CRTDUPOBJ, CRTLIB, CRTOUTQ, CRTSAVF, DLTF, DMPOBJ, DMPSYSOBJ, DSPCTLD, DSPDEVD, DSPHDWRSC, DSPJOB, DSPLIND, DSPLOG, DSPNWID, DSPPTF, DSPSFWRSC, OVRPRTF, PRTERRLOG, PRTINTDTA, SAVDLO, SAVLIB, SAVOJB, WRKACTJOB, and WRKSYSVAL	Use	
	Existing problem [7]		Change
SNDPTFORD (Q)			
SNDSRVRQS (Q)			
STRCMNTRC [3] (Q)	NWID (network ID) or line description	Use	
STRCPYSCN	Job queue	Use	
	Device description	Use	
	Output file, if specified	See General Rules on page D-2	See General Rules on page D-2
STRSRVJOB (Q)	User profile of job	Use	
STRSST [3] (Q)			
TRCCPIC (Q)			
TRCICF (Q)			
TRCINT (Q)			
TRCJOB (Q)	Output file, if specified	See General Rules on page D-2	See General Rules on page D-2
	Exit program, if specified	Use	

Service

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied profiles are authorized to the command. The security officer can grant Use authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
VFYCMN (Q)	Line description [5]	Use	
	Controller description [5]	Use	
	Network ID [5]	Use	
VFYLNKLPDA (Q)	Line description	Read	
VFYPRT (Q)	Device description	Use	
VFYTAP (Q)	Device description	Use	
WRKCNTINF (Q)			
WRKFSTAF (Q)	QUSRSYS/QPVINDEX *USRIDX	Change	Use
WRKFSTPCT (Q)	QUSRSYS/QPVPCTABLE *USRIDX	Change	Use
WRKPRB [1] (Q)	Line, controller, and device based on problem analysis action	Use and add	
WRKSRVPVD (Q)			

[1] You need authority to the PRTERRLOG command for some analysis procedures or if the error log records are being saved.

[2] All restrictions for the RSTOBJ command also apply.

[3] Service (*SERVICE) special authority is required to run this command.

[4] The objects listed are used by the command, but authority to the objects is not checked. Authority to use the command is sufficient to use the objects.

[5] You need use authority to the communications object you are verifying.

[6] You must have *SPLCTL special authority to save a spooled file.

[7] When SAVAPARDTA is run for a new problem, a unique APAR library is created for that problem. If you run SAVAPARDTA again for the same problem to collect more information, you must have Use authority to the APAR library for the problem.

[8] The option to add a new member to an existing output file is not valid for this command.

Spelling Aid Dictionaries

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CRTSPADCT	Spelling aid dictionary	Existence	
	Dictionary - REPLACE(*NO)		Add
	Dictionary - REPLACE(*YES)	See General Rules on page D-2	Add
DLTSPADCT	Spelling aid dictionary	Existence	
WRKSPADCT	Spelling aid dictionary	Operational	Use

Sphere of Control

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDSOCE	Sphere of control [1]	Use and add	
DSPSOCSTS			
RMVSOCE	Sphere of control [1]	Use and delete	
WRKSOC	Sphere of control [1]	Use	

[1] The sphere of control is physical file QUSRSYS/QAALSOC.

Spooled Files

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)	Output Queue Parameters			Special Authority
				DSPDTA	AUTCHK	OPRCTL	
CHGSPLFA [1,2]	Output queue [3]	Read, add, delete			*DTAAUT		
		Owner [4]			*OWNER		
						*YES	*JOBCTL
CHGSPLFA [1], if moving spooled file	Original output queue [3]	Read, add, delete			*DTAAUT		
		Owner [4]			*OWNER		
				*YES or *NO		*YES	*JOBCTL
	Spooled file	Owner		*OWNER			
	Target output queue [3]	Read					
						*YES	*JOBCTL
CPYSPLF [1]	Database file	See General Rules on page D-2	See General Rules on page D-2				
	Spooled file	Owner		*OWNER			
	Output queue [3]	Read		*YES			
		Read, add, delete		*NO	*DTAAUT		
		Owner [4]		*NO	*OWNER		
				*YES or *NO		*YES	*JOBCTL
DLTSPLF [1]	Output queue [3]	Read, add, delete			*DTAAUT		
		Owner [4]			*OWNER		
						*YES	*JOBCTL
DSPSPLF [1]	Output queue [3]	Read		*YES			
		Read, add, delete		*NO	*DTAAUT		
		Owner [4]		*NO	*OWNER		
				*YES or *NO		*YES	*JOBCTL
	Spooled file	Owner		*OWNER			
HLDSPLF [1]	Output queue [3]	Read, add, delete			*DTAAUT		
		Owner [4]			*OWNER		
						*YES	*JOBCTL
RCLSPLSTG (Q)							
RLSSPLF [1]	Output queue [3]	Read, add, delete			*DTAAUT		
		Owner [4]			*OWNER		
						*YES	*JOBCTL

Spooled Files

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)	Output Queue Parameters			Special Authority
				DSPDTA	AUTCHK	OPRCTL	
SNDNETSPLF [1,5]	Output queue [3]	Read		*YES			
		Read, add, delete		*NO	*DTAAUT		
		Owner [4]		*NO	*OWNER		
				*YES or *NO		*YES	*JOBCTL
	Spooled file	Owner		*OWNER			
WRKSPLF							
WRKSPLFA							

[1] Users are always authorized to control their own spooled files.

[2] To move a spooled file to the front of an output queue (PRTSEQ(*NEXT)) or change its priority to a value greater than the limit specified in your user profile, you must have one of the authorities shown for the output queue or have *SPLCTL special authority.

[3] If you have *SPLCTL special authority, you do not need any authority to the output queue.

[4] You must be the owner of the output queue.

[5] You must have *USE authority to the recipient's output queue and output queue library when sending a file to a user on the same system.

Subsystem Descriptions

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDAJE	Subsystem description	Operational and management	
	Job description	Operational	
ADDCMNE	Subsystem description	Operational and management	
	Job description	Operational	
	User profile	Operational	
ADDJOBQE	Subsystem description	Operational and management	
ADDPJE	Subsystem description	Operational and management	
	User profile for the program start request to specify *PGMSTRRQS	Use	
	User profile	Use	
	Job description	Operational	
ADDRTGE	Subsystem description	Operational and management	
ADDWSE	Subsystem description	Operational and management	
	Job description	Operational	
CHGAJE	Subsystem description	Operational and management	
	Job description	Operational	

Subsystem Descriptions

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHGCMNE	Subsystem description	Operational and management	
	Job description	Operational	
	User profile	Operational	
CHGJOBQE	Subsystem description	Operational and management	
CHGPJE	Subsystem description	Operational and management	
	User profile for the program start request to specify *PGMSTRRQS	Use	
	User profile	Use	
	Job description	Operational	
CHGRTGE	Subsystem description	Operational and management	
CHGSBSD	Subsystem description	Operational and management	
CHGWSE	Subsystem description	Operational and management	
	Job description	Operational	
CRTSBSD	Subsystem description		Add
DLTSBSD	Subsystem description	Operational and existence	
DSPSBSD	Subsystem description	Operational	
ENDSBS [1]			
ENDSYS [1]			
PWRDWNSYS [1]			
RMVAJE	Subsystem description	Operational and management	
RMVCMNE	Subsystem description	Operational and management	
RMVJOBQE	Subsystem description	Operational and management	
RMVPJE	Subsystem description	Operational and management	
RMVRTGE	Subsystem description	Operational and management	
RMVWSE	Subsystem description	Operational and management	
STRSBS	Subsystem description	Operational	
WRKSBS			
WRKSBSD	Subsystem description	Operational and management	Use

[1] You must have job control (*JOBCTL) special authority to use this command.

System

These commands do not require any object authorities:

CHGSHRPOOL
DSPSYSSTS
RCLACTGRP
RCLRSC
RETURN
RTVGRPA
SIGNOFF
WRKDSKSTS
WRKSHRPOOL
WRKSYSSTS

System Reply List

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDRPYLE (Q)			
CHGRPYLE (Q)			
RMVRPYLE (Q)			
WRKRPYLE			

System Values

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHGSYSVAL (Q) 1,2			
DSPSYSVAL			
RTVSYSVAL			
WRKSYSVAL 1,2			

1 To change some system values, you must have *ALLOBJ and *SECADM special authority.

2 To change some system values, you must have *AUDIT special authority.

System/36 Environment

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHGS36	S/36 configuration object QS36ENV	Update	
CHGS36A	S/36 configuration object QS36ENV	Update	
CHGS36PGMA	Program	Management and use	
CHGS36PRCA	File QS36PRC	Management and use	
CHGS36SRCA	Source	Management and use	
CRTMSGFMNU	Menu: REPLACE(*NO)		Add
	Menu: REPLACE(*YES)	See General Rules on page D-2	Add
	Display file if it exists	All	
	Message file	Use	Change
	Source file QS36SRC	All	
CRTS36DSPF	Display file: REPLACE(*NO)		Add
	Display file: REPLACE(*YES)	See General Rules on page D-2	Add, change
	To-file source file when TOMBR is not *NONE	All	Change
	Source file QS36SRC	Use	
	Create Display File (CRTDSPF) command	Operational	

System/36 Environment

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CRTS36MNU	Menu: REPLACE(*NO)		Add, change
	Menu: REPLACE(*YES)	See General Rules on page D-2	Add, change
	To-file source file when TOMBR is not *NONE	All	Change
	Source file QS36SRC	Use	
	Display file when REPLACE(*YES) is specified	All	
	Message files named in source	All	
	Display file		Change
	CRTMSGF command	Operational and existence	
	ADDMSGD command	Operational	
	CRTDSPF command	Operational	
CRTS36MSGF	Message file: REPLACE(*NO)		Add, change
	Message file: REPLACE(*YES)	See General Rules on page D-2	Add, change
	To-file source file when TOMBR is not *NONE	All	Change
	Source file QS36SRC	Use	
	Display file when REPLACE(*YES) is specified	All	
	Message file named in source	All	
	Message file named in source when OPTION is *ADD or *CHANGE	Change	
	Message files named in source when OPTION(*CREATE) is specified	All	
	CRTMSGF command	Operational and existence	
	ADDMSGD command	Operational	
	CHGMSGD command when OPTION(*CHANGE) is specified	Operational	
DSPS36	S/36 configuration object QS36ENV	Read	
EDTS36PGMA	Program, to modify attributes	Management and use	
	Program, to view attributes	Use	
EDTS36PRCA	File QS36PRC, to modify attributes	Management and use	
	File QS36PRC, to view attributes	Use	
EDTS36SRCA	Source file QS36SRC, to modify attributes	Management and use	
	Source file QS36SRC, to view attributes	Use	
RSTS36F	From-file	Use	
	To-file	All	See General Rules on page D-2
	Based-on physical file, if file being restored is a logical (alternative) file	Change	
	Device file or device description	Use	
RSTS36FLR [1,2,3]	S/36 folder	Use	
	To-folder	Change	
	Device file or device description	Use	
RSTS36LIBM	From-file	Use	
	To-file	All	See General Rules on page D-2
	Device file or device description	Use	
RTVS36A	S/36 configuration object QS36ENV	Update	

System/36 Environment

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
SAVS36F	From-file	Use	
	To-file, when it is a physical file	All	See General Rules on page D-2
	Device file or device description	Use	
SAVS36LIBM	From-file	Use	
	To-file, when it is a physical file	All	See General Rules on page D-2
	Device file or device description	Use	
WRKS36	S/36 configuration object QS36ENV	Read	
WRKS36PGMA	Program, to modify attributes	Management and use	
	Program, to view attributes	Use	
WRKS36PRCA	File QS36PRC, to modify attributes	Management and use	
	File QS36PRC, to view attributes	Use	
WRKS36SRCA	Source file QS36SRC, to modify attributes	Management and use	
	Source file QS36SRC, to view attributes	Use	

[1] You need *ALL authority to the document if replacing it. You need operational and all the data authorities to the folder if restoring new information into the folders, or you need *ALLOBJ special authority.

[2] If used for a data dictionary, the only the authority to the command is required.

[3] You must be enrolled in the system distribution directory if the source folder is a document folder.

Tables

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CRTTBL	Table		Add
DLTTBL	Table	Existence	
WRKTBL	Table	Operational	Use

Transmission Control Protocol/Internet Protocol Commands

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDTCPLNK	Line description	Use	
ADDTCPPORT			
ADDTCPRSI			
ADDTCPRTE			
CFGTCP			
CHGTCPA			
CHGTCPLNK	Line description	Use	
CHGTCPRTE			
CHGVTMAP			
CHGVT1MAP			
DSPVTMAP			
DSPVT1MAP			
ENDTCPCNN			
ENDTCPLNK			
FTP	Table objects	Use	

Transmission Control Protocol/Internet Protocol Commands

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
NETSTAT			
PING			
RMVTCPLNK			
RMVTCPPORT			
RMVTCPRSI			
RMVTCPRTE			
SETVTMAP			
SETVT1MAP			
SETVTTBL	Table objects	Use	
SNDTCPSPLF [1]	Workstation customizing object	Use	
STRTCPFTP	Table objects	Use	
STRTCPLNK			
STRTCPTELN	Table objects	Use	
TELNET	Table objects	Use	
VFYTCPCNN			
WRKNAMSMTP			
WRKTCPSTS			

[1] The SNDTCPSPLF command uses the same combinations of authorities, special authorities, and output queue parameters as the SNDNETSPLF command. See page D-59.

Token Ring

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDTRAINF			
CHGTRAINF			
DSPTRAPRF			
DSPTRNSTS			
RMVTRA			
RMVTRAINF			
WRKTRA			

Upgrade Order Information Data

*These commands are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
RMVACTTRA	Program QLMADRMV	Use	
WRKORDINF	QGPL/QMAHFILE file	All	

User Index, User Queue, User Space

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
DLTUSRIDX	User index	Existence	
DLTUSRQ	User queue	Existence	
DLTUSRSPC	User space	Existence	

User Profiles

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CHGDSTPWD [1]			
CHGPRF	User profile	Use and management	
	Initial program [2]	Use	
	Initial menu [2]	Use	
	Job description [2]	Use	
	Message queue [2]	Use	
	Output queue [2]	Use	
	Attention-key- handling program [2]	Use	
	Current library [2]	Use	
CHGPWD			
CHGUSRPRF [3]	User profile	Use and management	
	Initial program [2]	Use	
	Initial menu [2]	Use	
	Job description [2]	Use	
	Message queue [2]	Use	
	Output queue [2]	Use	
	Attention-key-handling program [2]	Use	
	Current library [2]	Use	
	Group profile [2,4]	Change and management	
CHKPWD			
CRTUSRPRF [3]	Initial program	Use	
	Initial menu	Use	
	Job description	Use	
	Message queue	Use	
	Output queue	Use	
	Attention-key-handling program	Use	
	Current library	Use	
	Group profile [4]	Change and management	
DLTUSRPRF [3,9]	User profile	Use and existence	
	Message queue [5]	Existence, use, and delete	
DSPAUTUSR [6]	User profile	Read	
	Output file	See General Rules on page D-2	See General Rules on page D-2
DSPPGMADP	User profile	Management	
	Output file	See General Rules on page D-2	See General Rules on page D-2
DSPUSRPRF	User profile	Read	
	Output file	See General Rules on page D-2	See General Rules on page D-2
GRTUSRAUT [7]	Referenced user profile	Read	
	Objects you are granting authority to	Management	
RSTAUT (Q) [8]			
RSTUSRPRF (Q) [8,10]			

User Profiles

*Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C shows which IBM-supplied profiles are authorized to the command. The security officer can grant *USE authority to others.*

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
RTVUSRPRF	User profile	Read	
SAVSECDTA [8]	Save file, if empty	Use and add	
	Save file, if records exist	Use, add, and management	
WRKUSRPRF	User profile	Read	Use

[1] This command can be run only if you are signed on as QSECOFR.

[2] You need authority only to the objects for fields you are changing in the user profile.

[3] *SECADM special authority is required.

[4] Authority to the group profile cannot come from adopted authority.

[5] The message queue associated with the user profile is deleted if it is owned by that user profile. To delete the message queue, the user running the DLTUSRPRF command must have the authorities specified.

[6] The display includes only user profiles to which the user running the command has the specified authority.

[7] See the authorities required for the GRTOBJAUT command on page D-3.

[8] *SAVSYS special authority is required.

[9] If you select the option to delete objects owned by the user profile, you must have the necessary authority for the delete operations. If you select the option to transfer ownership to another user profile, you must have the necessary authority to the objects and to the target user profile. See information for the CHGOBJOWN command on page D-3.

[10] You must have *ALLOBJ special authority to specify ALWOBJDIF(*ALL).

Workstation Customizing

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CRTWSCST	Source file	Use	
	Workstation customizing object, if REPLACE(*NO)		Add
	Workstation customizing object, if REPLACE(*YES)	Existence and management	Add
DLTWSCST	Workstation customizing object	Existence	
RTVWSCST	To-file, if it exists and a new member is added	Operational, management, and add	
	To-file, if file and member exist	Operational, add, and delete	
	To-file, if the file does not exist		Add

Writers

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)	Output Queue Parameters		Special Authority
				AUTCHK	OPRCTL	
CHGWTR [1,2]	Output queue	Read, add, delete		*DTAAUT		
		Owner [3]		*OWNER		
					*YES	*JOBCTL
ENDWTR [1]	Output queue	Read, add, delete		*DTAAUT		
		Owner [3]		*OWNER		
					*YES	*JOBCTL

Writers

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)	Output Queue Parameters		Special Authority
				AUTCHK	OPRCTL	
HLDWTR [1]	Output queue	Read, add, delete		*DTAAUT		
		Owner [3]		*OWNER		
					*YES	*JOBCTL
RLSWTR [1]	Output queue	Read, add, delete		*DTAAUT		
		Owner [3]		*OWNER		
					*YES	*JOBCTL
STRDKTWTR [1]	Output queue	Read, add, delete		*DTAAUT		
		Owner [3]		*OWNER		
					*YES	*JOBCTL
	Message queue	Operational and add				
	Device description	Read				
STRPRTWTR [1]	Output queue	Read, add, delete		*DTAAUT		
		Owner [3]		*OWNER		
					*YES	*JOBCTL
	Message queue	Operational and add				
	Device description	Read				
WRKWTR						

[1] If you have *SPLCTL special authority, you do need any authority to the output queue.

[2] To change the output queue for the writer, you need one of the specified authorities for the new output queue.

[3] You must be the owner of the output queue.

Appendix E. Security APIs and Authority for Call Level Interfaces

Table E-1 gives a brief description of the application programming interfaces (APIs) available to access security information on the system.

The *System Programmer's Interface Reference* provides complete information about using call-level interfaces.

Attempts by *USER state programs to use call level interfaces that are not listed here causes an AF entry to be written to the audit journal, if the auditing function is active. At security level 40 and higher, attempts to use unsupported call-level interfaces fail. See "Preventing the Use of Unsupported Interfaces" on page 2-5 for more information.

Table E-1. Security Application Programming Interfaces (APIs)

API Name	API Description	Program Name	Default Public Authority	Similar Commands
Change Previous Sign-On Date	Changes the date last signed on in the user profile for the current process.	QSYCHGPR	*USE	None
Change User Password	Changes a user's password.	QSYCHGPW	*USE	CHGPWD
Check User Authority to an Object	Returns an indication about a user's specified authority to an object.	QSYCUSRA	*USE	CHKOBJ
Check User Special Authorities	Returns an indication of a user's special authorities.	QSYCUSRS	*USE	None
Convert Authority Values to MI Value	Converts authority values to the machine interface (MI) representation of the value.	QSYCVTA	*USE	None
Get Profile Handle	Validates a user ID and password, and creates an encrypted abbreviation called a profile handle for that user profile.	QSYGETPH	*EXCLUDE	None
List Authorized Users	Puts a list of authorized users of the system in a user space.	QSYLAUTU	*USE	DSPAUTUSR
List Objects Secured by Authorization List	Puts a list of objects secured by an authorization list in a user space.	QSYLATLO	*USE	DSPAUTLOBJ
List Objects That Adopt Owner Authority	Puts a list of objects that adopt an owner's authority in a user space.	QSYLOBJP	*USE	DSPPGMADP
List Objects User Is Authorized to or Owns	Puts a list of objects that a user owns or is authorized to in a user space.	QSYLOBJA	*USE	DSPUSRPRF
List Users Authorized to Object	Puts a list of users privately authorized to an object in a user space.	QSYLUSRA	*USE	DSPAUTL DSPOBJAUT
Release Profile Handle	Deletes a profile handle.	QSYRLSPH	*EXCLUDE	None
Retrieve Information about a User	Returns the information about a user.	QSYRUSRI	*USE	RTVUSRPRF DSPUSRPRF
Retrieve User Authority to Object	Returns the user's authority to an object.	QSYRUSRA	*USE	None
Set Profile	Switches the job to run under a new profile.	QWTSETP	*EXCLUDE	None

Appendix F. Layout of Audit Journal Entries

This appendix contains layout information for all entry types in the audit (QAUDJRN) journal. Table F-1 contains the layout for fields that are common to all entry types. This layout, called QORJDE2, is the default when you create an output file using the DSPJRN command.

Tables F-2 through F-36 contain layouts for the field reference files provided to define entry-specific data. You can use the CRTDUPOBJ command to create any empty output file with the same layout as one of the field reference files. You can use the DSPJRN command to copy selected entries from the audit journal to the output file for analysis. "Analyzing Audit Journal Entries with Query or a Program" on page 9-13 provides examples of using the field reference files.

*Table F-1. Standard Heading Fields for Audit Journal Entries. QJORDJE2 Record Format (*TYPE2)*

Offset	Field	Format	Description
1	Length of Entry	Zoned(5,0)	Total length of the journal entry including the entry length field.
6	Sequence Number	Zoned(10,0)	Applied to each journal entry. Initially set to 1 for each new or restored journal. Optionally, reset to 1 when a new receiver is attached.
16	Journal Code	Char(1)	Always T.
17	Entry Type	Char(2)	AD Auditing changes AF Authority failure AP Obtaining authority through a program that adopts owner authority CA Authority changes CD Command string audit CO Create object CP User profile changed, created, or restored DO Delete object DS DST security password reset JD Change to User parameter of a job description JS Actions that affect jobs ML Office services mail actions NA Network attribute changed OM Object move or rename OR Object restore OW Object ownership changed PA Program changed to adopt authority PO Printed output PS Profile swap PW Invalid password RA Authority change during restore RJ Restoring job description with user profile specified RO Change of object owner during restore RP Restoring adopted authority program RU Restoring user profile authority SD Changes to system distribution directory SE Subsystem routing entry changed SF Actions to spooled files SM System management changes ST Use of service tools SV System value changed YC DLO object accessed (change) YR DLO object accessed (read) ZC Object accessed (change) ZR Object accessed (read)
19	Date of Entry	Char(6)	The system date that the entry was made.
25	Time of Entry	Zoned(6,0)	The system time that the entry was made.
31	Name of Job	Char(10)	The name of the job that caused the entry to be generated.
41	User Name	Char(10)	The user profile name associated with the job[1].
51	Job Number	Zoned(6,0)	The job number.
57	Program Name	Char(10)	The name of the program that made the journal entry.
67	Object Name	Char(10)	Used for file journaling. Not used for audit journal entries.
77	Library Name	Char(10)	Used for file journaling. Not used for audit journal entries.
87	Member Name	Char(10)	Used for file journaling. Not used for audit journal entries.
97	Count/RRN	Zoned(10)	Used for file journaling. Not used for audit journal entries.
107	Flag	Char(1)	Used for file journaling. Not used for audit journal entries.
108	Commit Cycle	Zoned(10)	Used for file journaling. Not used for audit journal entries.
118	User Profile	Char(10)	The name of the current user profile[1].
128	System Name	Char(8)	The name of the system.
136	(Reserved Area)	Char(20)	

[1] The three fields beginning at offset 31 make up the system job name. In most cases, the *User name* field at offset 41 and the *User profile name* field at offset 118 have the same value. For prestarted jobs, the *User profile name* field contains the name of the user starting the transaction. For some jobs, both these fields contain QSYS as the user name. The *User profile name* field in the entry-specific data contains the actual user who caused the entry. If an API is used to swap user profiles, the *User profile name* field contains the name of the new (swapped) user profile.

Table F-2. AD (Auditing Change) Journal Entries. QASYADJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 for field listing.
156	Entry Type	Char(1)	D CHGDLOAUD command O CHGOBJAUD command U CHGUSRAUD command
157	Object Name	Char(10)	Name of the object for which auditing was changed.
167	Library Name	Char(10)	Name of the library for the object.
177	Object Type	Char(8)	The type of object.
185	Object Audit Value	Char(10)	The new value specified on the CHGOBJAUD command.
195	CHGUSRAUD *CMD	Char(1)	Y = Audit commands for this user.
196	CHGUSRAUD *CREATE	Char(1)	Y = Write an audit record when this user creates an object.
197	CHGUSRAUD *DELETE	Char(1)	Y = Write an audit record when this user deletes an object.
198	CHGUSRAUD *JOBDTA	Char(1)	Y = Write an audit record when this user changes a job.
199	CHGUSRAUD *OBJMGT	Char(1)	Y = Write an audit record when this user moves or renames an object.
200	CHGUSRAUD *OFCSRV	Char(1)	Y = Write an audit record when this user performs office functions.
201	CHGUSRAUD *PGMADP	Char(1)	Y = Write an audit record when this user obtains authority through adopted authority.
202	CHGUSRAUD *SAVRST	Char(1)	Y = Write an audit record when this user saves or restores objects.
203	CHGUSRAUD *SECURITY	Char(1)	Y = Write an audit record when this user performs security-relevant actions.
204	CHGUSRAUD *SERVICE	Char(1)	Y = Write an audit record when this user performs service functions.
205	CHGUSRAUD *SPLFDTA	Char(1)	Y = Write an audit record when this user manipulates spooled files.
206	CHGUSRAUD *SYSMGT	Char(1)	Y = Write an audit record when this user makes system management changes.
207	Reserve Area	Char(20)	
227	DLO Name	Char(12)	Name of the DLO object for which auditing was changed.
239	(Reserved Area)	Char(8)	
247	Folder Path	Char(63)	Path of the folder.

Table F-3. AF (Authority Failure) Journal Entries. QASYAFJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Violation Type[1]	Char(1)	A Unauthorized object access attempt B Restricted instruction C Validation failure (see offset 185) D Use of unsupported interface, object domain failure J Submit job profile error P Profile swap error R Hardware protection error S Default sign-on attempt U User permission request not valid
157	Object Name	Char(10)	The name of the object.
167	Library Name	Char(10)	The name of the library the object is in.
177	Object Type	Char(8)	The type of object.
185	Validation Value Violation Type	Char(1)	The type of cyclic redundancy check (validation value), set only if the violation type (offset 156) is C. A Changed object was restored that may violate security. B Object restore and all authority revoked. C Validation Value failure on program. Copy of program that was translated was restored. D A changed object was restored as requested by the security officer. E System install time error detected.
186	Job Name	Char(10)	The name of the job.
196	User Name	Char(10)	The job user name.
206	Job Number	Zoned(6,0)	The job number.
212	Program Name	Char(10)	The name of the program.
222	Program Library	Char(10)	The name of the library where the program is found.
232	User Profile [2]	Char(10)	The name of the user using the program.
242	Work Station Name	Char(10)	The name of the work station or work station type.
252	Program Instruction Number	Zoned(7,0)	The instruction number of the program.
259	(Reserved Area)	Char(13)	
272	Office User	Char(10)	The name of the office user.
282	DLO Name	Char(12)	The name of the document library object.
294	(Reserved Area)	Char(8)	
302	Folder Path	Char(63)	The path of the folder.
365	Office on Behalf of User	Char(10)	User working on behalf of another user.

[1] For more information about the violation types, see Table 9-2 on page 9-7.

[2] This field contains the name of the user that caused the entry. The user at offsets 41 and 118 may be QSYS.

Table F-4. AP (Adopted Authority) Journal Entries. QASYAPJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	S Start E End
157	Object Name	Char(10)	The name of the program, service program, or SQL package
167	Library name	Char(10)	The name of the library.
177	Object Type	Char(8)	The type of object.
185	Owning User Profile	Char(10)	The name of the user profile whose authority is adopted.

Table F-5. CA (Authority Changes) Journal Entries. QASYCAJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry.
			A Changes to authority
157	Object Name	Char(10)	The name of the object.
167	Library Name	Char(10)	The name of the library the object is in.
177	Object Type	Char(8)	The type of object.
185	User Name	Char(10)	The name of the user profile whose authority is being granted or revoked.
195	Authorization List Name	Char(10)	The name of the authorization list.
			Authorities granted or removed:
205	Object Existence	Char(1)	Y *OBJEXIST
206	Object Management	Char(1)	Y *OBJMGT
207	Object Operational	Char(1)	Y *OBJOPR
208	Authorization List Management	Char(1)	Y *AUTLMGT
209	Authorization List	Char(1)	Y *AUTL public authority
210	Read Authority	Char(1)	Y *READ
211	Add Authority	Char(1)	Y *ADD
212	Update Authority	Char(1)	Y *UPD
213	Delete Authority	Char(1)	Y *DLT
214	Exclude Authority	Char(1)	Y *EXCLUDE
215	(Reserved Area)	Char(7)	
222	Command Type	Char(3)	The type of command used.
			GRT Grant RVK Revoke
225	(Reserved Area)	Char(20)	
245	Office User	Char(10)	The name of the office user.
255	DLO Name	Char(12)	The name of the DLO.
267	(Reserved Area)	Char(8)	
275	Folder Path	Char(63)	The path of the folder.
338	Office on Behalf of User	Char(10)	User working on behalf of another user.
348	Personal Status	Char(1)	Y Personal status changed
349	Access Code	Char(1)	A Access code added R Access code removed
350	Access Code	Char(4)	Access code.

Table F-6. CD (Command String) Journal Entries. QASYCDJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry.
			C Command run L OCL statement O Operator control command P S/36 procedure U Utility control statement
157	Object Name	Char(10)	The name of the object.
167	Library Name	Char(10)	The name of the library the object is in.
177	Object Type	Char(8)	The type of object.
185	Run from a CL program	Char(1)	Y Yes N No
186	Command string	Char(6000)	The command that was run, with parameters.

Table F-7. CO (Create Object) Journal Entries. QASYCOJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry. N Create of new object R Replacement of existing object
157	Object Name	Char(10)	The name of the object.
167	Library Name	Char(10)	The name of the library the object is in.
177	Object Type	Char(8)	The type of object.
185	(Reserved Area)	Char(20)	
205	Office User	Char(10)	The name of the office user.
215	DLO Name	Char(12)	The name of the document library object created.
227	(Reserved Area)	Char(8)	
235	Folder Path	Char(63)	The path of the folder.
298	Office on Behalf of User	Char(10)	User working on behalf of another user.

Table F-8. CP (User Profile Changes) Journal Entries. QASYCPJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry. A Change to a user profile
157	User Profile Name	Char(10)	The name of the user profile that was changed.
167	Library Name	Char(10)	The name of the library.
177	Object Type	Char(8)	The type of object.
185	Command Name	Char(3)	The type of command used. CRT CRTUSRPRF CHG CHGUSRPRF RST RSTUSRPRF DST QSECOFR password reset using DST
188	Password Changed	Char(1)	Y Password changed
189	Password *NONE	Char(1)	Y Password is *NONE.
190	Password Expired	Char(1)	Y Password expired
191	All Object Special Authority	Char(1)	Y *ALLOBJ special authority
192	Job Control Special Authority	Char(1)	Y *JOBCTL special authority
193	Save System Special Authority	Char(1)	Y *SAVSYS special authority
194	Security Administrator Special Authority	Char(1)	Y *SECADM special authority
195	Spool Control Special Authority	Char(1)	Y *SPLCTL special authority
196	Service Special Authority	Char(1)	Y *SERVICE special authority
197	Audit Special Authority	Char(1)	Y *AUDIT special authority
198	(Reserved Area)	Char(14)	
212	Group Profile	Char(10)	The name of a group profile.
222	Owner	Char(10)	Owner of objects created as a member of a group profile.
232	Group Authority	Char(10)	Group profile authority.
242	Initial Program	Char(10)	The name of the user's initial program.
252	Initial Program Library	Char(10)	The name of the library where the initial program is found.
262	Initial Menu	Char(10)	The name of the user's initial menu.
272	Initial Menu Library	Char(10)	The name of the library where the initial menu is found.
282	Current Library	Char(10)	The name of the user's current library.
292	Limited Capabilities	Char(10)	The value of limited capabilities parameter.
302	User Class	Char(10)	The user class of the user.
312	Priority Limit	Char(1)	The value of the priority limit parameter.
313	Profile Status	Char(10)	User profile status.

Table F-9. DO (Delete Operation) Journal Entries. QASYDOJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry.
			A Object was deleted
157	Object Name	Char(10)	The name of the object.
167	Library Name	Char(10)	The name of the library the object is in.
177	Object Type	Char(8)	The type of object.
185	(Reserved Area)	Char(20)	
205	Office User	Char(10)	The name of the office user.
215	DLO Name	Char(12)	The name of the document library object.
227	(Reserved Area)	Char(8)	
235	Folder Path	Char(63)	The path of the folder.
298	Office on Behalf of User	Char(10)	User working on behalf of another user.

Table F-10. DS (DST Password Reset) Journal Entries. QASYDSJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry.
			A Reset of DST password
157	DST Password Reset	Char(1)	Y Request to reset DST password.

Table F-11. JD (Job Description Change) Journal Entries. QASYJDJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry.
			A User profile specified for the USER parameter of a job description
157	Job Description	Char(10)	The name of the job description that had the USER parameter changed.
167	Library Name	Char(10)	The name of the library the object is in.
177	Object Type	Char(8)	The type of object.
185	Command Type	Char(3)	The type of command used.
			CHG Change Job Description (CHGJOBD) command. CRT Create Job Description (CRTJOBD) command.
188	Old User	Char(10)	The name of the user profile specified for the USER parameter before the job description was changed.
198	New User	Char(10)	The name of the user profile specified for the user parameter when the job description was changed.

Table F-12. JS (Job Change) Journal Entries. QASYJSJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry. A ENDJOBABN command B Submit C Change E End H Hold I Disconnect M Modify profile or group profile N ENDJOB command P Program start request attached to prestart job R Release S Start
157	Job Type	Char(1)	The type of job. A Autostart B Batch I Interactive M Subsystem monitor R Reader S System W Writer X SCPF
158	Job Subtype	Char(1)	The subtype of the job. ' ' No subtype D Batch immediate E Procedure start request J Prestart P Print driver T MRT U Alternate spool user
159	Job Name	Char(10)	The first part of the qualified job name being operated on
169	Job User Name	Char(10)	The second part of the qualified job name being operated on
179	Job Number	Char(6)	The third part of the qualified job name being operated on
185	Device Name	Char(10)	The name of the device
195	User Profile	Char(10)	The name of the user profile for the job
205	Job Description Name	Char(10)	The name of the job description for the job
215	Job Description Library	Char(10)	The name of the library for the job description
225	Job Queue Name	Char(10)	The name of the job queue for the job
235	Job Queue Library	Char(10)	The name of the library for the job queue
245	Output Queue Name	Char(10)	The name of the output queue for the job
255	Output Queue Library	Char(10)	The name of the library for the output queue
265	Printer Device	Char(10)	The name of the printer device for the job
275	Library List [1]	Char(430)	The library list for the job
705	Group Profile Name [1]	Char(10)	The name of the group profile for the job

[1] This field is blank if the job is on the job queue and has not run.

Table F-13. ML (Mail Actions) Journal Entries. QASYMLJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry. O Mail log opened
157	User Profile	Char(10)	User profile name.
167	User ID	Char(8)	User identifier
175	Address	Char(8)	User address

Table F-14. NA (Network Attribute Change) Journal Entries. QASYNAJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry. A Change to network attribute.
157	Network Attribute	Char(10)	The name of the network attribute.
167	New Network Attribute Value	Char(250)	The value of the network attribute after it was changed.
417	Old Network Attribute Value	Char(250)	The value of the network attribute before it was changed.

Table F-15. OM (Object Management Change) Journal Entries. QASYOMJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry. M Object moved to a different library. R Object renamed.
157	Old Object Name	Char(10)	The name of the old object.
167	Old Library Name	Char(10)	The name of the library the old object is in.
177	Object Type	Char(8)	The type of object.
185	New Object Name	Char(10)	The new name of the object.
195	New Library Name	Char(10)	The name of the library the object was moved to.
205	(Reserved Area)	Char(20)	
225	Office User	Char(10)	The name of the office user.
235	Old Folder or Document Name	Char(12)	The old name of the folder or document.
247	(Reserved Area)	Char(8)	
255	Old Folder Path	Char(63)	The old path of the folder.
318	New Folder or Document Name	Char(12)	The new name of the folder or document.
330	(Reserved Area)	Char(8)	
338	New Folder Path	Char(63)	The new path of the folder.
401	Office on Behalf of User	Char(10)	User working on behalf of another user.

Table F-16. OR (Object Restore) Journal Entries. QASYORJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry. N A new object was restored to the system. E An existing object was restored to the system.
157	Restored Object Name	Char(10)	The name of the restored object.
167	Restored Library Name	Char(10)	The name of the library of the restored object.
177	Object Type.	Char(8)	The type of object.
185	Save Object Name	Char(10)	The name of the save object.
195	Save Library Name	Char(10)	The name of the library from which the object was saved.
205	System State Program [1]	Char(1)	Y A system state program was restored. N A user state program was restored.
206	System Command [2]	Char(1)	Y A system command was restored. N A user state command was restored.
207	(Reserved Area)	Char(18)	
225	Office User	Char(10)	The name of the office user.
235	Restore DLO Name	Char(12)	The document library object name of the restored object.
247	(Reserved Area)	Char(8)	
255	Restore Folder Path	Char(63)	The folder into which the DLO was restored.
318	Save DLO Name	Char(12)	The DLO name of the saved object.
330	(Reserved Area)	Char(8)	
338	Save Folder Path	Char(63)	The folder from which the DLO was saved.
401	Office on Behalf of User	Char(10)	User working on behalf of another user.

[1] This field has an entry only if the object being restored is a program.
[2] This field has an entry only if the object being restored is a command.

Table F-17. OW (Ownership Change) Journal Entries. QASYOWJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry. A Change of object owner
157	Object Name	Char(10)	The name of the object.
167	Library Name	Char(10)	The name of the library the object is in.
177	Object Type	Char(8)	The type of object.
185	Old Owner	Char(10)	Old owner of the object.
195	New Owner	Char(10)	New owner of the object.
205	(Reserved Area)	Char(20)	
225	Office User	Char(10)	The name of the office user.
235	DLO Name	Char(12)	The name of the document library object.
247	(Reserved Area)	Char(8)	
255	Folder Path	Char(63)	The path of the folder.
318	Office on Behalf of User	Char(10)	User working on behalf of another user.

Table F-18. PA (Program Adopt) Journal Entries. QASYPAJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry. A Change program to adopt owner's authority
157	Program Name	Char(10)	The name of the program.
167	Program Library	Char(10)	The name of the library where the program is found.
177	Object Type	Char(8)	The type of object.
185	Owner	Char(10)	The name of the owner.

Table F-19. PO (Printer Output) Journal Entries. QASYPOJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Output Type	Char(1)	The type of output. D Direct print S Spooled file printed
157	Status After Printing	Char(1)	D Deleted after printed H Held after printed S Saved after printed ' ' Direct print
158	Job Name	Char(10)	The first part of the qualified job name.
168	Job User Name	Char(10)	The second part of the qualified job name.
178	Job Number	Zoned(6,0)	The third part of the qualified job name.
184	User Profile	Char(10)	The user profile that created the output.
194	Output Queue	Char(10)	The output queue containing the spooled file.[1]
204	Output Queue Library Name	Char(10)	The name of the library containing the output queue.[1]
214	Device Name	Char(10)	The device where the output was printed.
224	Device Type	Char(4)	The type of printer device.
228	Device Model	Char(4)	The model of the printer device.
232	Device File Name	Char(10)	The name of the device file used to access the printer.
242	Device File Library	Char(10)	The name of the library for the device file.
252	Spooled File Name	Char(10)	The name of the spooled file [1]
262	Spooled File Number	Char(4)	The number of the spooled file [1].
266	Form Type	Char(10)	The form type of the spooled file.
276	User Data	Char(10)	The user data associated with the spooled file [1].

[1] This field is blank if the type of output is direct print.

Table F-20. PS (Profile Swap) Journal Entries. QASYPSJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry. A Profile swap during pass-through. H Profile handle generated by the QSYGETPH API. S Start work on behalf of relationship E End work on behalf of relationship
157	User Profile	Char(10)	User profile name.
167	Source Location	Char(8)	Pass-through source location.
175	Original Target User Profile	Char(10)	Original pass-through target user profile.
185	New Target User Profile	Char(10)	New pass-through target user profile.
195	Office User	Char(10)	Office user starting or ending on behalf of relationship.
205	On Behalf of User	Char(10)	User on behalf of whom the office user is working.

Table F-21. PW (Password) Journal Entries. QASYPWJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Violation Entry Type	Char(1)	The type of violation P Password not valid U User name not valid A APPC bind failure
157	User Name	Char(10)	The job user name.
167	Device name	Char(40)	The name of the device or communications device on which the password or user ID was entered.
207	Remote Location Name	Char(8)	Name of the remote location for the APPC bind.
215	Local Location Name	Char(8)	Name of the local location for the APPC bind.
223	Network ID	Char(8)	Network ID for the APPC bind.

Table F-22. RA (Authority Change for Restored Object) Journal Entries. QASYRAJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry. A Changes to authority for object restored
157	Object Name	Char(10)	The name of the object.
167	Library Name	Char(10)	The name of the library the object is in.
177	Object Type	Char(8)	The type of object.
185	Authorization List Name	Char(10)	The name of the authorization list.
195	Public Authority	Char(1)	Y Public authority set to *EXCLUDE.
196	Private Authority	Char(1)	Y Private authority removed.
197	AUTL Removed	Char(1)	Y Authorization list removed from object.
198	(Reserved Area)	Char(20)	
218	DLO Name	Char(12)	The name of the document library object.
230	(Reserved Area)	Char(8)	
238	Folder Path	Char(63)	The folder containing the document library object.

Table F-23. RJ (Restoring Job Description) Journal Entries. QASYRJJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry. A Restoring a job description that had a user profile specified in the USER parameter.
157	Job Description Name	Char(10)	The name of the job description restored.
167	Library Name	Char(10)	The name of the library the job description was restored to.
177	Object Type	Char(8)	The type of object.
185	User Name	Char(10)	The name of the user profile specified in the job description.

Table F-24. RO (Ownership Change for Restored Object) Journal Entries. QASYROJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry.
			A Restoring objects that had ownership changed when restored
157	Object Name	Char(10)	The name of the object.
167	Library Name	Char(10)	The name of the library the object is in.
177	Object Type	Char(8)	The type of object.
185	Old Owner	Char(10)	The name of the owner before ownership was changed.
195	New Owner	Char(10)	The name of the owner after ownership was changed.
205	(Reserved Area)	Char(20)	
225	DLO Name	Char(12)	The name of the document library object.
237	(Reserved Area)	Char(8)	
245	Folder Path	Char(63)	The folder into which the object was restored.

Table F-25. RP (Restoring Programs that Adopt Authority) Journal Entries. QASYRPJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry.
			A Restoring programs that adopt the owner's authority
157	Program Name	Char(10)	The name of the program
167	Program Library	Char(10)	The name of the library in which the program is located
177	Object Type	Char(8)	The type of object
185	Owner Name	Char(10)	Name of the owner

Table F-26. RU (Restore Authority for User Profile) Journal Entries. QASYRUJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry.
			A Restoring authority to user profiles
157	User Name	Char(10)	The name of the user profile whose authority was restored.
167	Library Name	Char(10)	The name of the library.
177	Object Type	Char(8)	The type of object.

Table F-27. SD (Change System Distribution Directory) Journal Entries. QASYSDJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry.
			S System directory change
157	Type of Change	Char(3)	ADD Add directory entry CHG Change directory entry COL Collector entry DSP Display directory entry OUT Output file request PRT Print directory entry RMV Remove directory entry RNM Rename directory entry RTV Retrieve details SUP Supplier entry
160	Type of record	Char(4)	DIRE Directory DPTD Department details SHDW Directory shadow SRCH Directory search
164	Originating System	Char(8)	The system originating the change
172	User Profile	Char(10)	The user profile making the change
182	Requesting system	Char(8)	The system requesting the change
190	Function Requested	Char(6)	INIT Initialization OFFLIN Offline initialization REINIT Reinitialization SHADOW Normal shadowing STPSHD Stop shadowing
196	User ID	Char(8)	The user ID being changed
204	Address	Char(8)	The address being changed
212	Network User ID	Char(47)	The network user ID being changed

Table F-28. SE (Change of Subsystem Routing Entry) Journal Entries. QASYSEJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry.
			A Subsystem routing entry changed
157	Subsystem Name	Char(10)	The name of the object
167	Library Name	Char(10)	The name of the library the object is in
177	Object Type	Char(8)	The type of object.
185	Program Name	Char(10)	The name of the program that changed the routing entry
195	Library Name	Char(10)	The name of the library for the program
205	Sequence Number	Char(4)	The sequence number
209	Command Name	Char(3)	The type of command used
			ADD ADDRTGE CHG CHGRTGE RMV RMVRTGE

Table F-29. SF (Action to Spooled File) Journal Entries. QASYSFJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Access Type	Char(1)	The type of entry A Spooled file read C Spooled file created D Spooled file deleted H Spooled file held I Create of inline file R Spooled file released U Spooled file changed
157	Database File Name	Char(10)	The name of the database file containing the spooled file
167	Library Name	Char(10)	The name of the library for the database file
177	Object Type	Char(8)	The object type of the database file
185	Reserved area	Char(10)	
195	Member Name	Char(10)	The name of the file member.
205	Spooled File Name	Char(10)	The name of the spooled file [1].
215	Spooled File Number	Char(4)	The number of the spooled file [1].
219	Output Queue Name	Char(10)	The name of the output queue containing the spooled file.
229	Output Queue Library	Char(10)	The name of the library for the output queue.
239	Reserved area	Char(20)	
259	Old Copies	Char(3)	Number of old copies of the spooled file
262	New Copies	Char(3)	Number of new copies of the spooled file
265	Old Printer	Char(10)	Old printer for the spooled file
275	New Printer	Char(10)	New printer for the spooled file
285	New Output Queue	Char(10)	New output queue for the spooled file
295	New Output Queue Library	Char(10)	Library for the new output queue
305	Old Form Type	Char(10)	Old form type of the spooled file
315	New Form Type	Char(10)	New form type of the spooled file
325	Old Restart Page	Char(8)	Old restart page for the spooled file
333	New Restart Page	Char(8)	New restart page for the spooled file
341	Old Page Range Start	Char(8)	Old page range start of the spooled file
349	New Page Range Start	Char(8)	New page range start of the spooled file
357	Old Page Range End	Char(8)	Old page range end of the spooled file
365	New Page Range End	Char(8)	New page range end of the spooled file

[1] This field is blank when the type of entry is I (inline print).

Table F-30. SM (System Management Change) Journal Entries. QASYSMJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	Function accessed B Backup list changed C Automatic cleanup options D DRDA F HFS file system N Network file operation O Backup options changed P Power on/off schedule S System reply list
157	Access Type	Char(1)	A Add C Change D Delete R Remove S Display T Retrieve or receive
158	Sequence Number	Char(4)	Sequence number of the action
162	Message ID	Char(7)	Message ID associated with the action
169	Relational Database Name	Char(18)	Name of the relational database
187	File System Name	Char(10)	Name of the file system
197	Backup Option Changed	Char(10)	The backup option that was changed
207	Backup List Change	Char(10)	The name of the backup list that was changed
217	Network File Name	Char(10)	The name of the network file that was used
227	Network File Member	Char(10)	The name of the member of the network file
237	Network File Number	Zoned(6,0)	The number of the network file
243	Network File Owner	Char(10)	The name of the user profile that owns the network file
253	Network File Originating User	Char(8)	The name of the user profile that originated the network file
261	Network File Originating Address	Char(8)	The address that originated the network file

Table F-31. ST (Service Tools Action) Journal Entries. QASYSTJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry A Service record
157	Service Tool	Char(2)	CS STRCPYSCN DC DLTCMNTRC DD DMPDLO DO DMPOBJ DS DMPSYSOBJ EC ENDCMNTRC PC PRTCMNTRC PE PRTERRLOG PI PRTINTDTA SC STRCMNTRC SJ STRSRVJOB ST STRSST TI TRCINT
159	Object Name	Char(10)	Name of the object accessed
169	Library Name	Char(10)	Name of the library for the object
177	Object Type	Char(8)	Type of object
187	Job Name Being Traced	Char(10)	The first part of the qualified job name
197	Job User Name	Char(10)	The second part of the qualified job name
207	Job Number	Zoned(6,0)	The third part of the qualified job name
213	Object Name	Char(30)	Name of the object for DMPSYSOBJ
243	Library Name	Char(30)	Name of the library for the object for DMPSYSOBJ
273	Object Type	Char(8)	Type of the object
281	DLO Name	Char(12)	Name of the document library object
293	(Reserved Area)	Char(8)	
301	Folder Path	Char(63)	The folder containing the document library object

Table F-32. SV (Action to System Value) Journal Entries. QASYSVJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	The type of entry. A Change to system values
157	System Value	Char(10)	The name of the system value
167	New System Value	Char(250)	The value to which the system value was changed
417	Old System Value	Char(250)	The value of the system value before it was changed
667	New System Value Continued	Char(250)	Continuation of the value to which the system value was changed
917	Old System Value Continued	Char(250)	Continuation of the value of the system value before it was changed

Table F-33. YC (Change to DLO Object) Journal Entries. QASYYCJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	Object access
			C Change of a DLO object
157	Object Name	Char(10)	Name of the object
167	Library Name	Char(10)	Name of the library
177	Object Type	Char(8)	Type of object
185	Office User	Char(10)	User profile of the office user
195	Folder or Document Name	Char(12)	Name of the document or folder
207	(Reserved Area)	Char(8)	
215	Folder Path	Char(63)	The folder containing the document library object
278	On Behalf of User	Char(10)	User working on behalf of another user
288	Access Type	Packed(5,0)	Type of access [1]

[1] See Table F-37 on page F-20 for a list of the codes for access types.

Table F-34. YR (Read of DLO Object) Journal Entries. QASYYRJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	Object access
			R Read of a DLO object
157	Object Name	Char(10)	Name of the object
167	Library Name	Char(10)	Name of the library
177	Object Type	Char(8)	Type of object
185	Office User	Char(10)	User profile of the office user
195	Folder or Document Name	Char(12)	Name of the document library object
207	(Reserved Area)	Char(8)	
215	Folder Path	Char(63)	The folder containing the document library object
278	On Behalf of User	Char(10)	User working on behalf of another user
288	Access Type	Packed(5,0)	Type of access [1]

[1] See Table F-37 on page F-20 for a list of the codes for access types.

Table F-35. ZC (Change to Object) Journal Entries. QASYZCJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	Object access
			C Change of an object
157	Object Name	Char(10)	Name of the object
167	Library Name	Char(10)	Name of the library in which the object is located
177	Object Type	Char(8)	Type of object
185	Access Type	Packed(5,0)	Type of access [1]
188	Access Specific Data	Char(50)	Specific data about the access

[1] See Table F-37 on page F-20 for a list of the codes for access types.

Table F-36. ZR (Read of Object) Journal Entries. QASYZRJE Field Description File

Offset	Field	Format	Description
1			Heading fields common to all entry types. See Table F-1 on page F-2 for field listing.
156	Entry Type	Char(1)	Object access
			R Read of an object
157	Object Name	Char(10)	Name of the object
167	Library Name	Char(10)	Name of the library in which the object is located
177	Object Type	Char(8)	Type of object
185	Access Type	Packed(5,0)	Type of access [1]
188	Access Specific Data	Char(50)	Specific data about the access

[1] See Table F-37 on page F-20 for a list of the codes for access types.

Table F-37 lists the access codes used for object auditing journal entries in files QASYYCJE, QASYYRJE, QASYZCJE, and QASYZRJE.

Table F-37. Numeric Codes for Access Types

Code	Access Type	Code	Access Type	Code	Access Type
1	Add	22	File	43	Retrieve
2	Activate Program	23	Grant	44	Run
3	Analyze	24	Hold	45	Revoke
4	Apply	25	Initialize	46	Save
5	Call or TFRCTL	26	Load	47	Save with Storage Free
6	Configure	27	List	48	Save and Delete
7	Change	28	Move	49	Submit
8	Check	29	Merge	50	Set
9	Close	30	Open	51	Send
10	Clear	31	Print	52	Start
11	Compare	32	Query	53	Transfer
12	Cancel	33	Reclaim	54	Trace
13	Copy	34	Receive	55	Verify
14	Create	35	Read	56	Vary
15	Convert	36	Reorganize	57	Work
16	Debug	37	Release	58	Read/Change DLO Attribute
17	Delete	38	Remove	59	Read/Change DLO Security
18	Dump	39	Rename	60	Read/Change DLO Content
19	Display	40	Replace	61	Read/Change DLO all parts
20	Edit	41	Resume		
21	End	42	Restore		

Appendix G. Object Operations and Auditing

This appendix lists operations that can be performed against objects on the system, and whether those operations are audited. The lists are organized by object type. The operations are grouped by whether they are audited when *ALL or *CHANGE is specified for the OBJAUD value of the CHGOBJAUD or CHGDLOAUD command.

Whether an audit record is written for an action depends on a combination of system values, a value in the user profile of the user performing the action, and a value defined for the object. “Planning the Auditing of Object Access” on page 9-9 describes how to set up auditing for objects.

Operations shown in the tables in uppercase, such as CPYF, refer to CL commands, unless they are labeled as an application programming interface (API).

Operations Common to All Object Types:

- Read operation

CRTDUPOBJ	Create Duplicate Object (if *ALL is specified for *from-object*)
DMPOBJ	Dump Object
DMPSYSOBJ	Dump System Object
SAVCHGOBJ	Save Changed Object
SAVLIB	Save Library
SAVOBJ	Save Object
SAVSAVFDTA	Save Save File Data
SAVDLO	Save DLO Object
SAVLICPGM	Save Licensed Program

Note: The audit record for the save operation will identify if the save was done with the STG(*FREE).

- Change operation

CHGOBJD	Change Object Description
CHGOBJOWN	Change Object Owner
CRTxxxxxx	Create object

Notes:

1. If *ALL or *CHANGE is specified for the target library, a ZC entry is written when an object is created.
2. If *CREATE is active for action auditing, a CO entry is written when an object is created.

DLTxxxxxx	Delete object

Notes:

1. If *ALL or *CHANGE is specified for the library containing the object, a ZC entry is written when an object is deleted.
2. If *ALL or *CHANGE is specified for the object, a ZC entry is written when it is deleted.
3. If *DELETE is active for action auditing, a DO entry is written when an object is deleted.

GRTOBJAUT	Grant Object Authority

Note: If authority is granted based on a referenced object, an audit record is not written for the referenced object.

MOVOBJ	Move Object
RCLSTG	Reclaim Storage: – If an object is secured by a damaged *AUTL, an audit record is written when the object is secured by the QRCLAUTL authorization list. – An audit record is written if an object is moved into the QRCL library.
RNMOBJ	Rename Object
RSTCFG	Restore Configuration Objects
RSTLIB	Restore Library
RSTLICPGM	Restore Licensed Program
RSTOBJ	Restore Object
RVKOBJAUT	Revoke Object Authority

- Operations that are not audited

Prompt [1]	Prompt override program for a change command (if one exists)
CHKOBJ	Check Object
ALCOBJ	Allocate Object
CPROBJ	Compress Object
DCPOBJ	Decompress Object
DLCOBJ	Deallocate Object
DSPOBJD	Display Object Description
DSPOBJAUT	Display Object Authority
EDTOBJAUT	Edit Object Authority

Note: If object authority is changed and action auditing includes *SECURITY, or the object is being audited, an audit record is written.

[1] A prompt override program displays the current values when prompting is requested for a command. For example, if you type CHGURSPRF USERA and press F4 (prompt), the Change User Profile display shows the current values for the USERA user profile.

QSYCUSRA	Check User's Authority to an Object API
QSYLUSRA	List Users Authorized to an Object API. An audit record is not written for the object whose authority is being listed. An audit record is written for the user space used to contain information.
QSYRUSRA	Retrieve User's Authority to Object API
RCLTMPSTG	Reclaim Temporary Storage
RTVOBJD	Retrieve Object Description
SAVSTG	Save Storage (audit of SAVSTG command only)
WRKOBJLCK	Work with Object Lock
WRKOBJOWN	Work with Objects by Owner
WRKxxx	Work with object commands

*Operations for Alert Table (*ALRTBL):*

- Read operation

None

- Change operation

ADDALRD	Add Alert Description
CHGALRD	Change Alert Description
CHGALRTBL	Change Alert Table
RMVALRD	Remove Alert Description

- Operations that are not audited

Print	Print alert description
WRKALRD	Work with Alert Description
WRKALRTBL	Work with Alert Table

*Operations for Authorization List (*AUTL):*

- Read operation

None

- Change operation

ADDAUTLE	Add Authorization List Entry
CHGAUTLE	Change Authorization List Entry
EDTAUTL	Edit Authorization List
RMVAUTLE	Remove Authorization List Entry

- Operations that are not audited

DSPAUTL	Display Authorization List
DSPAUTLOBJ	Display Authorization List Objects
DSPAUTLDLO	Display Authorization List DLO
RTVAUTLE	Retrieve Authorization List Entry
QSYLATLO	List Objects Secured by *AUTL API
WRKAUTL	Work with authorization list

*Operations for Authority Holder (*AUTHLR):*

- Read operation

None

- Change operation

Associated	When used to secure an object.

- Operations that are not audited

DSPAUTHLR	Display Authority Holder

*Operations for Binding Directory (*BNDDIR):*

- Read operation

CRTPGM	Create Program
CRTSRVPGM	Create Service Program

- Change operation

ADDBNDDIRE	Add Binding Directory Entries
RMVBNDDIRE	Remove Binding Directory Entries

- Operations that are not audited

DSPBNDDIR	Display the contents of a binding directory
WRKBNDDIR	Work with Binding Directory
WRKBNDDIRE	Work with Binding Directory Entry

*Operations for Configuration List (*CFGL):*

- Read operation

CPYCFGL	Copy Configuration List. An entry is written for the *from-configuration-list*

- Change operation

ADDCFGLE	Add Configuration List Entries
CHGCFGL	Change Configuration List
CHGCFGLE	Change Configuration List Entry
RMVCFGLE	Remove Configuration List Entry

- Operations that are not audited

DSPCFGL	Display Configuration List
WRKCFGL	Work with Configuration List

*Operations for Chart Format (*CHTFMT):*

- Read operation

Display	DSPCHT command or option F10 from the BGU menu
Print/Plot	DSPCHT command or option F15 from the BGU menu
Save/Create	Save or create graphics data file (GDF) using CRTGDF command or option F13 from the BGU menu

- Change operation

None

- Operations that are not audited

None

*Operations for C Locale Description (*CLD):*

- Read operation

RTVCLDSRC	Retrieve C Locale Source
Setlocale	Use the C locale object during C program run time using the Set locale function.

- Change operation

None

- Operations that are not audited

None

Operations for Class (*CLS):

- Read operation

  None

- Change operation

CHGCLS	Change Class

- Operations that are not audited

Job start	When used by work management to start a job
DSPCLS	Display Class
WRKCLS	Work with Class

Operations for Command (*CMD):

- Read operation

Run	When command is run

- Change operation

CHGCMD	Change Command
CHGCMDDFT	Change Command Default

- Operations that are not audited

DSPCMD	Display Command
PRTCMDUSG	Print Command Usage
QCDRCMDI	Retrieve Command Information API
WRKCMD	Work with Command

  The following commands are used within CL programs to control processing and to manipulate data within the program. Their use is not audited.

  CALL [1]
  CHGVAR
  DCL
  DCLF
  DO
  ELSE
  ENDDO
  ENDPGM
  ENDRCV
  GOTO
  IF
  MONMSG
  PGM
  RCVF
  RETURN
  SNDF
  SNDRCVF
  TRFCTL
  WAIT

  [1] CALL is audited if it is run interactively. It is not audited if it is run within a CL program.

Operations for Connection List (*CNNL):

- Read operation

  None

- Change operation

ADDCNNLE	Add Connection List Entry
CHGCNNL	Change Connection List
CHGCNNLE	Change Connection List Entry
RMVCNNLE	Remove Connection List Entry
RNMCNNLE	Rename Connection List Entry

- Operations that are not audited

Copy	Option 3 of WRKCNNL
DSPCNNL	Display Connection List
RTVCFGSRC	Retrieve source of connection list
WRKCNNL	Work with Connection List
WRKCNNLE	Work with Connection List Entry

Operations for Class-of-Service Description (*COSD):

- Read operation

  None

- Change operation

CHGCOSD	Change Class-of-Service Description

- Operations that are not audited

DSPCOSD	Display Class-of-Service Description
RTVCFGSRC	Retrieve source of class-of-service description
WRKCOSD	Copy class-of-service description
WRKCOSD	Work with Class-of-Service Description

Operations for Communications Side Information (*CSI):

- Read operation

DSPCSI	Display Communications Side Information
Initialize	Initialize conversation

- Change operation

CHGCSI	Change Communications Side Information

- Operations that are not audited

WRKCSI	Work with Communications Side Information

Operations for Cross System Product Map (*CSPMAP):

- Read operation

Reference	When referred to in a CSP application

- Change operation

  None

- Operations that are not audited

DSPCSPOBJ	Display CSP Object
WRKOBJCSP	Work with Objects for CSP

Operations for Cross System Product Table (*CSPTBL):

- Read operation

Reference	When referred to in a CSP application

- Change operation

  None

- Operations that are not audited

DSPCSPOBJ	Display CSP Object
WRKOBJCSP	Work with Objects for CSP

Operations for Controller Description (*CTLD):

- Read operation

VFYCMN	Link test
VRYCFG	Vary controller description on or off

- Change operation

CHGCTLxxx	Change controller description

- Operations that are not audited

DSPCTLD	Display Controller Description
PRTDEVADR	Print Device Address
RTVCFGSRC	Retrieve source of controller description
RTVCFGSTS	Retrieve controller description status
WRKCTLD	Copy controller description
WRKCTLD	Work with Controller Description

Operations for Device Description (*DEVD):

- Read operation

Acquire	First acquire of the device during open operation
Allocate	Allocate conversation
STRPASTHR	Start pass-through session Start of the second session for intermediate pass-through
VFYCMN	Link test
VRYCFG	Vary device description on or off

- Change operation

CHGDEVxxx	Change device description
HLDDEVxxx	Hold device description
RLSDEVxxx	Release device description
QWSSETWS	Change type-ahead setting for a device

- Operations that are not audited

DSPDEVD	Display Device Description
DSPMODSTS	Display Mode Status
RTVCFGSRC	Retrieve source of device description
RTVCFGSTS	Retrieve device description status
WRKCFGSTS	Work with device status
WRKDEVD	Copy device description
WRKDEVD	Work with Device Description

Operations for Directory Services:

Note: Directory services actions are audited if the action auditing (QAUDLVL) system value or the action auditing (AUDLVL) parameter in the user profile includes *OFCSRV.

- Operations that are audited

Add	Adding new directory entries
Change	Changing directory entry details
Delete	Deleting directory entries
Rename	Renaming directory entries
Print	Displaying or printing directory entry details Displaying or printing department details Displaying or printing directory entries as the result of a search
RTVDIRE	Retrieve Directory Entry
Collect	Collecting directory entry data using directory shadowing
Supply	Supplying directory entry data using directory shadowing

- Operations that are not audited

CL commands	CL commands that work on the directory may be audited separately using the object auditing function. **Note:** Some CL directory commands cause an audit record because they perform a function that is audited by *OFCSRV action auditing, such as adding a directory entry.
CHGDIRA	Change Directory Attributes
Departments	Adding, changing, deleting, or displaying directory department data
Descriptions	Assigning a description to a different directory entry using option 8 from the WRKDIR panel. Adding, changing, or deleting directory entry descriptions.
Distribution lists	Adding, changing, or deleting distribution lists
ENDDIRSHD	End Directory Shadowing
List	Displaying or printing a list of directory entries that does not include directory entry details, such as using the WRKDIR command or using F4 to select entries for sending a note.
Locations	Adding, changing, deleting, or displaying directory location data
Nickname	Adding, changing or deleting nicknames
Search	Searching for directory entries
STRDIRSHD	Start Directory Shadowing

Operations for Document Library Object (*DOC or *FLR):

- Read operation

CHKDOC	Check document spelling
CPYDOC	Copy Document
DMPDLO	Dump DLO
DSPDLOAUD	Display DLO Auditing **Note:** If auditing information is displayed for all documents in a folder, and object auditing is specified for the folder, an audit record is written. Displaying object auditing for individual documents does not result in an audit record.
DSPDLOAUT	Display DLO Authority
DSPDOC	Display Document
DSPHLPDOC	Display Help Document
EDTDLOAUT	Edit DLO Authority
MRGDOC	Merge Document
PRTDOC	Print Document
QHFCPYSF	Copy Stream File API
QHFGETSZ	Get Stream File Size API
QHFRDDR	Read Directory Entry API
QHFRDSF	Read Stream File API
RTVDOC	Retrieve Document
SAVDLO	Save DLO
SNDDOC	Send Document

SNDDST	Send Distribution
WRKDOC	Work with Document

Note: A read entry is written for the folder containing the documents.

- Change operation

ADDDLOAUT	Add DLO Authority
ADDOFCENR	Add Office Enrollment
CHGDLOAUD	Change DLO Auditing
CHGDLOAUT	Change DLO Authority
CHGDLOOWN	Change DLO Ownership
CHGDOCD	Change Document Description
CHGDSTD	Change Distribution Description
CPYDOC [2]	Copy Document

Note: A change entry is written if the target document already exists.

CRTFLR	Create Folder
CVTTOFLR [2]	Convert to Folder
DLTDLO [2]	Delete DLO
DTLDOCL [2]	Delete Document List
DLTDST [2]	Delete Distribution
EDTDLOAUT	Edit DLO Authority
EDTDOC	Edit Document
FILDOC [2]	File Document
GRTACCAUT	Grant Access Code Authority
GRTUSRPMN	Grant User Permission
MOVDOC [2]	Move Document
MRGDOC [2]	Merge Document
PAGDOC	Paginate Document
QHFCHGAT	Change Directory Entry Attributes API
QHFSETSZ	Set Stream File Size API
QHFWRTSF	Write Stream File API
QRYDOCLIB [2]	Query Document Library

Note: A change entry is written if an existing document resulting from a search is replaced.

RCVDST [2]	Receive Distribution
RGZDLO	Reorganize DLO
RMVACC	Remove access code, for any DLO to which the access code is attached
RMVDLOAUT	Remove DLO authority
RNMDLO [2]	Rename DLO
RPLDOC	Replace Document
RSTDLO [2]	Restore DLO
RTVDOC	Retrieve Document (check out)
RVKACCAUT	Revoke Access Code Authority
RVKUSRPMN	Revoke User Permission
SAVDLO [2]	Save DLO

- Operations that are not audited

ADDACC	Add Access Code
DSPACC	Display Access Code
DSPUSRPMN	Display User Permission
QHFCHGFP	Change File Pointer API
QHFCLODR	Close Directory API
QHFCLOSF	Close Stream File API
QHFFRCSF	Force Buffered Data API
QHFLULSF	Lock/Unlock Stream File Range API
QHFRTVAT	Retrieve Directory Entry Attributes API
RCLDLO	Reclaim DLO (*ALL or *INT)
WRKDOCLIB	Work with Document Library
WRKDOCPRTQ	Work with Document Print Queue

Operations for Data Area (*DTAARA):

- Read operation

DSPDTAARA	Display Data Area
RCVDTAARA	Receive Data Area (S/38 command)
RTVDTAARA	Retrieve Data Area
QWCRDTAA	Retrieve Data Area API

- Change operation

CHGDTAARA	Change Data Area
SNDDTAARA	Send Data Area

- Operations that are not audited

Data Areas	Local Data Area, Group Data Area, PIP (Program Initialization Parameter) Data Area
WRKDTAARA	Work with Data Area

Operations for Interactive Data Definition Utility (*DTADCT):

- Read operation

None

- Change operation

Create	Data dictionary and data definitions
Change	Data dictionary and data definitions
Copy	Data definitions (recorded as create)
Delete	Data dictionary and data definitions
Rename	Data definitions

- Operations that are not audited

Display	Data dictionary and data definitions
LNKDTADFN	Linking and unlinking file definitions
Print	Data dictionary, data definitions, and where-used information for data definitions

Operations for Data Queue (*DTAQ):

- Read operation

None

- Change operation

QRCVDTAQ	Receive Data Queue
QSNDDTAQ	Send Data Queue
QCLRDTAQ	Clear Data Queue

- Operations that are not audited

WRKDTAQ	Work with Data Queue

[2] A change entry is written if the target of the operation is in a folder.

Operations for Edit Description (*EDTD):

- Read operation

DSPEDTD	Display Edit Description
QECCVTEC	Edit code expansion (via routine QECEDITU)

- Change operation

None

- Operations that are not audited

WRKEDTD	Work with Edit Descriptions
QECEDT	Edit API
QECCVTEW	API for translating Edit Work into Edit Mask

Operations for Forms Control Table (*FCT):

- No Read or Change operations are audited for the *FCT object type.

Operations for File (*FILE):

- Read operation

CPYF	Copy File (uses open operation)
Open	Open of a file for read
DSPPFM	Display Physical File Member (uses open operation)
Open	Open of MRTs after the initial open
CRTBSCF	Create BSC File (uses open operation)
CRTCMNF	Create Communications File (uses open operation)
CRTDSPF	Create Display File (uses open operation)
CRTICFF	Create ICF File (uses open operation)
CRTMXDF	Create MXD File (uses open operation)
CRTPRTF	Create Printer File (uses open operation)
CRTPF	Create Physical File (uses open operation)
CRTLF	Create Logical File (uses open operation)
DSPMODSRC	Display Module Source (uses open operation)
STRDBG	Start Debug (uses open operation)
QTEDBGS	Retrieve View Text API

- Change operation

Open	Open a file for modification
CPYF	Copy File (open file for modification, such as adding records, clearing a member, or saving a member
ADDBSCDEVE	(S/38E) Add Bisync Device Entry to a mixed device file
ADDCMNDEVE	(S/38E) Add Communications Device Entry to a mixed device file
ADDDSPDEVE	(S/38E) Add Display Device Entry to a mixed device file
ADDICFDEVE	(S/38E) Add ICF Device Entry to a mixed device file
ADDLFM	Add Logical File Member
ADDPFM	Add Physical File Member
ADDPFVLM	Add Physical File Variable Length Member
APYJRNCHG	Apply Journaled Changes (one entry per file member changed)
CHGBSCF	Change Bisync function
CHGCMNF	(S/38E) Change Communications File
CHGDDMF	Change DDM File
CHGDKTF	Change Diskette File
CHGDSPF	Change Display File
CHGICFDEVE	Change ICF Device File Entry
CHGICFF	Change ICF File
CHGMXDF	(S/38E) Change Mixed Device File
CHGLF	Change Logical File
CHGLFM	Change Logical File Member
CHGPF	Change Physical File
CHGPFM	Change Physical File Member
CHGPRTF	Change Printer Device GQle
CHGSAVF	Change Save File
CHGTAPF	Change Tape Device File
CLRPFM	Clear Physical File Member
ENDJRNAP	End Journal Access Path (entry per file)
ENDJRNPF	End Journal Physical File (entry per file)
INZPFM	Initialize Physical File Member
JRNAP	(S/38E) Start Journal Access Path (entry per file)
JRNPF	(S/38E) Start Journal Physical File (entry per file)
RGZPFM	Reorganize Physical File Member
RMVBSCDEVE	(S/38E) Remove BSC Device Entry from a mixed dev file
RMVCMNDEVE	(S/38E) Remove CMN Device Entry from a mixed dev file
RMVDSPDEVE	(S/38E) Remove DSP Device Entry from a mixed dev file
RMVICFDEVE	(S/38E) Remove ICF Device Entry from an ICM dev file
RMVJRNCHG	Remove Journaled Changes (one entry per file member changed)
RMVM	Remove Member
RNMM	Rename Member
STRJRNAP	Start Journal Access Path (entry per file)
STRJRNPF	Start Journal Physical File (entry per file)
CHGS36PRCA	Change S/36 Procedure Attributes
EDTS36PRCA	Edit S/36 Procedure Attributes
WRKS36PRCA	Work with S/36 Procedure Attributes
CHGS36SRCA	Change S/36 Source Attributes
WRKS36SRCA	Work with S/36 Source Attributes
EDTS36SRCA	Edit S/36 Source Attributes

- Operations that are not audited

DSPFD	Display File Description
DSPFFD	Display File Field Description
DSPDBR	Display Database Relations
DSPPGMREF	Display Program File References
OVRxxx	Override file
RTVMBRD	Retrieve Member Description

WRKF	Work with File

Operations for Folder (*FLR)

- See operations for Document Library Object (*DOC or *FLR)

Operations for Font Resource (*FNTRSC):

- Read operation

Print	Referring to the font resource when creating a spooled file

- Change operation

  None

- Operations that are not audited

WRKFNTRSC	Work with Font Resource
Print	Printing a spooled file that refers to the font resource

Operations for Form Definition (*FORMDF):

- Read operation

Print	Referring to the form definition when creating a spooled file

- Change operation

  None

- Operations that are not audited

WRKFORMDF	Work with Form Definition
Print	Printing a spooled file that refers to the form definition

Operations for Filter Object (*FTR):

- Read operation

  None

- Change operation

ADDALRACNE	Add Alert Action Entry
ADDALRSLTE	Add Alert Selection Entry
CHGALRACNE	Change Alert Action Entry
CHGALRSLTE	Change Alert Selection Entry
CHGFTR	Change Filter
RMVFTRACNE	Remove Alert Action Entry
RMVFTRSLTE	Remove Alert Selection Entry
WRKFTRACNE	Work with Alert Action Entry
WRKFTRSLTE	Work with Alert Selection Entry

- Operations that are not audited

WRKFTR	Work with Filter
WRKFTRACNE	Work with Filter Action Entries
WRKFTRSLTE	Work with Filter Selection Entries

Operations for Graphics Symbols Set (*GSS):

- Read operation

Loaded	When it is loaded
Font	When it is used as a font in an externally described printer file

- Change operation

  None.

- Operations that are not audited

WRKGSS	Work with Graphic Symbol Set

Operations for Double-Byte Character Set Dictionary (*IGCDCT):

- Read operation

DSPIGCDCT	Display IGC Dictionary

- Change operation

EDTIGCDCT	Edit IGC Dictionary

Operations for Double-Byte Character Set Sort (*IGCSRT):

- Read operation

CPYIGCSRT	Copy IGC Sort (*from-*ICGSRT-object*)

- Change operation

CPYIGCSRT	Copy IGC Sort (*to-*ICGSRT-object*)

Operations for Double-Byte Character Set Table (*IGCTBL):

- Read operation

CPYIGCTBL	Copy IGC Table
STRFMA	Start Font Management Aid

- Change operation

STRFMA	Start Font Management Aid

- Operations that are not audited

CHKIGCTBL	Check IGC Table

Operations for Job Description (*JOBD):

- Read operation

  None

- Change operation

CHGJOBD	Change Job Description

- Operations that are not audited

DSPJOBD	Display Job Description
WRKJOBD	Work with Job Description
QWDRJOBD	Retrieve Job Description API
Batch job	When used to establish a job

Operations for Job Queue (*JOBQ):

- Read operation

  None

- Change operation

Entry	When an entry is placed on or removed from the queue
CLRJOBQ	Clear Job Queue
HLDJOBQ	Hold Job Queue
RLSJOBQ	Release Job Queue

- Operations that are not audited

ADDJOBQE [3]	Add Job Queue Entry
CHGJOB	Change Job from one JOBQ to another JOBQ
CHGJOBQE [3]	Change Job Queue Entry
QSPRJOBQ	Retrieve job queue information
RMVJOBQE [3]	Remove Job Queue Entry
TFRJOB	Transfer Job
TFRBCHJOB	Transfer Batch Job
WRKJOBQ	Work with Job Queue for a specific job queue
WRKJOBQ	Work with Job Queue for all job queues

Operations for Job Scheduler Object (*JOBSCD):

- Read operation

  None

- Change operation

ADDJOBSCDE	Add Job Schedule Entry
CHGJOBSCDE	Change Job Schedule Entry
RMVJOBSCDE	Remove Job Schedule Entry
HLDJOBSCDE	Hold Job Schedule Entry
RLSJOBSCDE	Release Job Schedule Entry

- Operations that are not audited

Display	Display details of scheduled job entry
WRKJOBSCDE	Work with Job Schedule Entries
Work with ...	Work with previously submitted jobs from job schedule entry
QWCLSCDE	List job schedule entry API

Operations for Journal (*JRN):

- Read operation

CMPJRNIMG	Compare Journal Images
DSPJRN	Display Journal Entry
RCVJRNE	Receive journal entry
RTVJRNE	Retrieve journal entry

- Change operation

APYJRNCHG	Apply Journaled Changes
CHGJRN	Change Journal
ENDJRNAP	End Journal Access Path
ENDJRNPF	End Journal Physical File
JRNAP	(S/38E) Start Journal Access Path
JRNPF	(S/38E) Start Journal Physical File
RMVJRNCHG	Remove Journaled Changes
SNDJRNE	Send Journal Entry (user entries only via SNDJRNE command)
STRJRNAP	Start Journal Access Path
STRJRNPF	Start Journal Physical File

- Operations that are not audited

WRKJRN	Work with Journal (DSPJRNMNU in S/38 environment)
WRKJRNA	Work with Journal Attributes (DSPJRNA in S/38 environment)

Operations for Journal Receiver (*JRNRCV):

- Read operation

  None

- Change operation

CHGJRN	Change Journal (when attaching new receivers)

- Operations that are not audited

DSPJRNRCVA	Display Journal Receiver Attributes
WRKJRNRCV	Work with Journal Receiver

Operations for Library (*LIB):

- Read operation

  DSPLIB — Display Library (when not empty. If library is empty, no audit is performed.)

  Locate — When a library is accessed to find an object

  Notes:

  1. Several audit entries may be written for a library for a single command. For example, when you open a file, a ZR audit journal entry for the library is written when the system locates the file and each member in the file.
  2. No audit entry is written if the locate function is not successful. For example, you run a command using a generic parameter, such as:

     ```
     DSPOBJD OBJECT(AR*/WRK*) +
        OBJTYPE(*FILE)
     ```

     If a library whose name begins with "AR" does not have any file names beginning with "WRK," no audit record is written for that library.

- Change operation

Library list	Adding library to a library list
CHGLIB	Change Library
CLRLIB	Clear Library
MOVOBJ	Move Object
RNMOBJ	Rename Object
Add	Add object to library
Delete	Delete object from library

- Operations that are not audited

  None

[3] An audit record is written if object auditing is specified for the subsystem description (*SBSD).

*Operations for Line Description (*LIND):*

- Read operation

VRYCFG	Vary on/off line description
RUNLPDA	Run LPDA-2 operational commands
VFYCMN	Link test
VFYLNKLPDA	LPDA-2 link test

- Change operation

CHGLINxxx	Change Line Description

- Operations that are not audited

Copy	Option 3 from WRKLIND
DSPLIND	Display Line Description
RTVCFGSRC	Retrieve Source of line description
RTVCFGSTS	Retrieve line description status
WRKLIND	Work with Line Description
WRKCFGSTS	Work with line description status

Operations for Mail Services:

Note: Mail services actions are audited if the action auditing (QAUDLVL) system value or the action auditing (AUDLVL) parameter in the user profile includes *OFCSRV.

- Operations that are audited

Change	Changes to the system distribution directory
On behalf	Working on behalf of another user **Note:** Working on behalf of another user is audited if the AUDLVL in the user profile or the QAUDLVL system value includes *SECURITY.
Open	An audit record is written when the mail log is opened

- Operations that are not audited

Change	Change details of a mail item
Delete	Delete a mail item
File	File a mail item into a document or folder **Note:** When a mail item is filed, it becomes a document library object (DLO). Object auditing can be specified for a DLO.
Forward	Forward a mail item
Print	Print a mail item **Note:** Printing of mail items can be audited using the *SPLFDTA or *PRTDTA audit level.
Receive	Receive a mail item
Reply	Reply to a mail item
Send	Send a mail item
View	View a mail item

*Operations for Menu (*MENU):*

- Read operation

Display	Displaying a menu through the GO MENU command or UIM dialog command

- Change operation

CHGMNU	Change Menu

- Operations that are not audited

Return	Returning to a menu in the menu stack that has already been displayed
DSPMNUA	Display Menu Attributes
WRKMNU	Work with Menu

*Operations for Mode Description (*MODD):*

- Read operation

None

- Change operation

CHGMODD	Change Mode Description

- Operations that are not audited

DSPMODD	Display Mode Description
WRKMODD	Work with Mode Descriptions

*Operations for Module Object (*MODULE):*

- Read operation

CRTPGM	An audit entry for each module object used during a CRTPGM.
CRTSRVPGM	An audit entry for each module object used during a CRTSRVPGM

- Change operation

CHGMOD	Change Module

- Operations that are not audited

DSPMOD	Display Module
WRKMOD	Work with Module

*Operations for Message File (*MSGF):*

- Read operation

DSPMSGD	Display Message Description
MRGMSGF	Merge Message File from-file
Print	Print message description
RTVMSG	Retrieve information from a message file
WRKMSGD	Work with Message Description

- Change operation

MRGMSGF	Merge Message File (to-file and replace MSGF)
ADDMSGD	Add Message Description
CHGMSGD	Change Message Description
RMVMSGD	Remove Message Description

- Operations that are not audited

OVRMSGF	Override Message File
WRKMSGF	Work with Message File

*Operations for Message Queue (*MSGQ):*

- Read operation

DSPLOG	Display Log
DSPMSG	Display Message

RCVMSG	Receive Message RMV(*NO)

- Change operation

CHGMSGQ	Change Message Queue
CLRMSGQ	Clear Message Queue
RCVMSG	Receive Message RMV(*YES)
RMVMSG	Remove Message
SNDxxxMSG	Send a Message to a message queue
SNDRPY	Send Reply
WRKMSG	Work with Message

- Operations that are not audited

WRKMSGQ	Work with Message Queue
Program	Program message queue operations

*Operations for Node List (*NODL):*

- Read operation

QFVLSTNL	List node list entries

- Change operation

ADDNODLE	Add Node List Entry
RMVNODLE	Remove Node List Entry

- Operations that are not audited

WRKNODL	Work with Node List
WRKNODLE	Work with Node List Entries

*Operations for Network Identifier (*NWID):*

- Read operation

VRYCFG	Vary network interface description on or off

- Change operation

CHGNWIISDN	Change Network Interface Description

- Operations that are not audited

Copy	Option 3 of WRKNWID
DSPNWID	Display Network Interface Description
RTVCFGSRC	Retrieve Source of Network Interface Description
RTVCFGSTS	Retrieve Status of Network Interface Description
WRKNWID	Work with Network Interface Description
WRKCFGSTS	Work with network interface description status

*Operations for Output Queue (*OUTQ):*

- Read operation

STRPRTWTR	Start a Printer Writer to an OUTQ

- Change operation

Placement	When an entry is placed on or removed from the queue
CHGOUTQ	Change Output Queue
CHGSPLFA [4]	Change Spooled File Attributes, if moved to a different output queue and either output queue is audited
CLROUTQ	Clear Output Queue
DLTSPLF [4]	Delete Spooled File
HLDOUTQ	Hold Output Queue
RLSOUTQ	Release Output Queue

- Operations that are not audited

CHGSPLFA [4]	Change Spooled File Attributes
CPYSPLF [4]	Copy Spooled File
Create [4]	Create a spooled file
DSPSPLF [4]	Display Spooled File
HLDSPLF [4]	Hold Spooled File
QSPROUTQ	Retrieve output queue information
RLSSPLF [4]	Release Spooled File
SNDNETSPLF [4]	Send Network Spooled File
WRKOUTQ	Work with Output Queue
WRKOUTQD	Work with Output Queue Description
WRKSPLF	Work with Spooled File
WRKSPLFA	Work with Spooled File Attributes

*Operations for Overlay (*OVL):*

- Read operation

Print	Referring to the overlay when creating a spooled file

- Change operation

None

- Operations that are not audited

WRKOVL	Work with overlay
Print	Printing a spooled file that refers to the overlay

*Operations for Page Definition (*PAGDFN):*

- Read operation

Print	Referring to the form definition when creating a spooled file

- Change operation

None

- Operations that are not audited

WRKPAGDFN	Work with Page Definition
Print	Printing a spooled file that refers to the page definition

*Operations for Page Segment (*PAGSEG):*

- Read operation

Print	Referring to the page segment when creating a spooled file

- Change operation

[4] This is also audited if action auditing (QAUDLVL system value or AUDLVL user profile value) includes *SPLFDTA.

None

- Operations that are not audited

WRKPAGSEG	Work with Page Segment
Print	Printing a spooled file that refers to the page segment

Operations for Print Descriptor Group (*PDG):

- Read operation

Open	When the page descriptor group is opened for read access by a PrintManager API or CPI verb.

- Change operation

Open	When the page descriptor group is opened for change access by a PrintManager API or CPI verb.

- Operations that are not audited

CHGPDGPRF	Change Print Descriptor Group Profile
WRKPDG	Work with Print Descriptor Group

Operations for Program (*PGM):

- Read operation

Activation	Program activation
Call	Call program that is not already activated
ADDPGM	Add program to debug
QTEDBGS	Qte Register Debug View API
QTEDBGS	Qte Retrieve Module Views API
// RUN	Run program in S/36 environment
RTVCLSRC	Retrieve CL Source
STRDBG	Start Debug

- Change operation

CHGCSPPGM	Change CSP/AE Program
CHGPGM	Change Program
CHGS36PGMA	Change S/36 Program Attributes
EDTS36PGMA	Edit S/36 Program Attributes
WRKS36PGMA	Work with S/36 Program Attributes

- Operations that are not audited

ANZPGM	Analyze Program
DMPCLPGM	Dump CL Program
DSPCSPOBJ	Display CSP Object
DSPPGM	Display Program
PRTCMDUSG	Print Command Usage
PRTCSPAPP	Print CSP Application
QBNLPGMI	List ILE Program Information API
QCLRPGMI	Retrieve Program Information API
STRCSP	Start CSP Utilities
TRCCSP	Trace CSP Application
WRKOBJCSP	Work with Objects for CSP
WRKPGM	Work with Program

Operations for Panel Group (*PNLGRP):

- Read operation

ADDSCHIDXE	Add Search Index Entry
QUIOPNDA	Open Panel Group for Display API
QUIOPNPA	Open Panel Group for Print API
QUHDSPH	Display Help API

- Change operation

None

- Operations that are not audited

WRKPNLGRP	Work with Panel Group

Operations for Product Availability (*PRDAVL):

- Change operation

WRKSPTPRD	Work with Supported Products, when support is added or removed

- Operations that are not audited

Read	No read operations are audited

Operations for Product Definition (*PRDDFN):

- Change operation

ADDPRDLICI	Add Product License Information
WRKSPTPRD	Work with Supported Products, when support is added or removed

- Operations that are not audited

Read	No read operations are audited

Operations for Product Load (*PRDLOD):

- Change operation

Change	Product load state, product load library list, product load folder list, primary language

- Operations that are not audited

Read	No read operations are audited

Operations for Query Manager Form (*QMFORM):

- Read operation

STRQMQRY	Start Query Management Query
RTVQMFORM	Retrieve Query Management Form
Run	Run a query
Export	Export a Query Management form
Print	Print a Query Management form Print a Query Management report using the form
Use	Access the form using option 2, 5, 6, or 9 or function F13 from the SQL/400 Query Manager menu.

- Change operation

CRTQMFORM	Create Query Management Form
IMPORT	Import Query Management form
Save	Save the form using a menu option or a command
Copy	Option 3 from the Work with QM Forms function

- Operations that are not audited

WRKQMFORM	Work with Query Management Forms

Active	Any form operation that is done against the 'active' form.

Operations for Query Manager Query (*QMQRY):

- Read operation

RTVQMQRY	Retrieve Query Manager Query
Run	Run Query Manager Query
STRQMQRY	Start Query Manager Query
Export	Export Query Manager query
Print	Print Query Manager query
Use	Access the query using function F13 or option 2, 5, 6, or 9 from the Work with Query Manager queries function

- Change operation

  None

- Operations that are not audited

Work with	When *QMQRYs are listed in a Work with display
Active	Any query operation that is done against the 'active' query.

Operations for Query Definition (*QRYDFN):

- Read operation

ANZQRY	Analyze Query
Change	Change a query using a prompt display presented by WRKQRY, QRY, or OfficeVision/400.
Display	Display a query using WRKQRY prompt display
Export	Export form using Query Manager
Export	Export query using Query Manager
Merge	Data merge when editing an OfficeVision/400 document and doing a direct merge of Query/400 data
	Data merge when printing or using the MRGDOC command on an OfficeVision/400 document that contains multicopy merge, column list merge, or get data field heading instructions
Print	Print query definition using WRKQRY prompt display
	Print Query Management form
	Print Query Management query
	Print Query Management report
QRYRUN	Run Query
Read	Reference a query by editing an OfficeVision/400 document and using the Query/400 licensed program to create column list merge or multicopy merge instructions
RTVQMFORM	Retrieve Query Management Form
RTVQMQRY	Retrieve Query Management Query
Run	Run query using WRKQRY prompt display
	Run (Query Management command)
RUNQRY	Run Query
STRQMQRY	Start Query Management Query
Submit	Submit a query (run request) to batch using WRKQRY prompt display or Exit This Query prompt display

- Change operation

Change	Save a changed query using the Query/400 licensed program

- Operations that are not audited

Run	Run a query using option 1 on the "Exit this Query" display when creating or changing a query using the Query/400 licensed program; Run a query interactively using PF5 while creating, displaying, or changing a query using the Query/400 licensed program
Merge	Merge query data using option 6 (direct merge), 7 (column list merge), or 8 (multicopy merge) on the Query/400 "Exist this Query" display following a create or change operation where the Query/400 licensed program is called from the OfficeVision/400 licensed program.

Operations for Reference Code Translate Table (*RCT):

- Read operation

  None

- Change operation

  None

- Operations that are not audited

  None

Operations for Reply List:

Note: Reply list actions are audited if the action auditing (QAUDLVL) system value or the action auditing (AUDLVL) parameter in the user profile includes *SYSMGT.

- Operations that are audited

ADDRPYLE	Add Reply List Entry
CHGRPYLE	Change Reply List Entry
RMVRPYLE	Remove Reply List Entry
WRKRPYLE	Work with Reply List Entry

- Operations that are not audited

  None

Operations for Subsystem Description (*SBSD):

- Read operation

ENDSBS	End Subsystem
STRSBS	Start Subsystem

- Change operation

ADDAJE	Add Autostart Job Entry
ADDCMNE	Add Communications Entry
ADDJOBQE	Add Job Queue Entry
ADDPJE	Add Prestart Job Entry

ADDRTGE	Add Routing Entry
ADDWSE	Add Workstation Entry
CHGAJE	Change Autostart Job Entry
CHGCMNE	Change Communications Entry
CHGJOBQE	Change Job Queue Entry
CHGPJE	Change Prestart Job Entry
CHGRTGE	Change Routing Entry
CHGSBSD	Change Subsystem Description
CHGWSE	Change Workstation Entry
RMVAJE	Remove Autostart Job Entry
RMVCMNE	Remove Communications Entry
RMVJOBQE	Remove Job Queue Entry
RMVPJE	Remove Prestart Job Entry
RMVRTGE	Remove Routing Entry
RMVWSE	Remove Workstation Entry

- Operations that are not audited

DSPSBSD	Display Subsystem Description
QWCLASBS	List Active Subsystem API
QWDLSJBQ	List Subsystem Job Queue API
QWDRSBSD	Retrieve Subsystem Description API
WRKSBSD	Work with Subsystem Description
WRKSBS	Work with Subsystem
WRKSBSJOB	Work with Subsystem Job

*Operations for Information Search Index (*SCHIDX):*

- Read operation

STRSCHIDX	Start Index Search
WRKSCHIDXE	Work with Search Index Entry

- Change operation (audited if OBJAUD is *CHANGE or *ALL)

ADDSCHIDXE	Add Search Index Entry
CHGSCHIDX	Change Search Index
RMVSCHIDXE	Remove Search Index Entry

- Operations that are not audited

WRKSCHIDX	Work with Search Index

*Operations for Spelling Aid Dictionary (*SPADCT):*

- Read operation

Verify	Spell verify function
Aid	Spell aid function
Hyphenation	Hyphenation function
Dehyphenation	Dehyphenation function
Synonyms	Synonym function
Base	Using dictionary as base when creating another dictionary
Verify	Using as verify dictionary when creating another dictionary
Retrieve	Retrieve Stop Word List Source
Print	Print Stop Word List Source

- Change operation

CRTSPADCT	Create Spelling Aid Dictionary with REPLACE(*YES)

- Operations that are not audited

None

Operations for Spooled Files:

Note: Spooled file actions are audited if the action auditing (QAUDLVL) system value or the action auditing (AUDLVL) parameter in the user profile includes *SPLFDTA.

- Operations that are audited

Access	Each access by any user that is not the owner of the spooled file, including: – CPYSPLF – DSPSPLF – SNDNETSPLF – SNDTCPSPLF – QSPOPNSP API
Change	Changing any of the following spooled file attributes: – COPIES – DEV – FORMTYPE – RESTART – PAGERANGE
Create	Creating a spooled file using print operations Creating a spooled file using the QSPCRTSP API
Delete	Deleting a spooled file using any of the following: – Printing a spooled file by a printer or diskette writer – Clearing the output queue (CLROUTQ) – Deleting the spooled file using the DLTSPLF command or the delete option from a spooled files display – Deleting spooled files when a job ends (ENDJOB SPLFILE(*YES)) – Deleting spooled files when a print job ends (ENDPJ SPLFILE(*YES))
Hold	Holding a spooled file by any of the following: – Using the HLDSPLF command – Using the hold option from a spooled files display – Printing a spooled file that specifies HOLD(*YES)
Read	Reading a spooled file by a printer or diskette writer
Release	Releasing a spooled file

*Operations for SQL Package (*SQLPKG):*

- Read operation

Run	When *SQLPKG object is run

- Change operation

None

- Operations that are not audited

None

Operations for Service Program (*SRVPGM):

- Read operation

CRTPGM	An audit entry for each service program used during a CRTPGM command
CRTSRVPGM	An audit entry for each service program used during a CRTSRVPGM command
QTEDBGS	Register Debug View API
QTEDBGS	Retrieve Module Views API

- Change operation

CHGSRVPGM	Change Service Program

- Operations that are not audited

DSPSRVPGM	Display Service Program
QBNLSPGM	List Service Program Information API
QBNRSPGM	Retrieve Service Program Information API
WRKSRVPGM	Work with Service Program

Operations for Session Description (*SSND):

- No Read or Change operations are audited for the *SSND object type.

Operations for S/36 Machine Description (*S36):

- Read operation

None

- Change operation

CHGS36	Change S/36 configuration
CHGS36A	Change S/36 configuration attributes
SET	SET procedure
CRTDEVXXX	When a device is added to the configuration table
DLTDEVD	When a device is deleted from the configuration table
RNMOBJ	Rename device description

- Operations that are not audited

DSPS36	Display S/36 configuration
RTVS36A	Retrieve S/36 Configuration Attributes
STRS36	Start S/36
ENDS36	End S/36

Operations for Table (*TBL):

- Read operation

QDCXLATE	Translate character string
QTBXLATE	Translate character string
QLGRTVSS	Retrieve sort sequence table
CRTLF	Translation Table during CRTLF command
Read	Use of Sort Sequence Table when running any command that can specify a sort sequence

- Change operation

None

- Operations that are not audited

WRKTBL	Work with table

Operations for User Index (*USRIDX):

- Read operation

QUSRTVUI	Retrieve user index entries API

- Change operation

QUSADDUI	Add User Index Entries API
QUSRMVUI	Remove User Index Entries API

- Operations that are not audited

Access	Direct access to a user index using MI instructions (only allowed for a user domain index in a library specified in the QALWUSRDMN system value.
QUSRUIAT	Retrieve User Index Attributes API

Operations for User Profile (*USRPRF):

- Read operation

None

- Change operation

CHGPRF	Change Profile
CHGPWD	Change Password
CHGUSRPRF	Change User Profile
DLTUSRPRF	Delete User Profile
GRTUSRAUT	Grant User Authority (*to-user-profile*)
QSYCHGPW	Change Password API
RSTUSRPRF	Restore User Profile

- Operations that are not audited

DSPUSRPRF	Display User Profile
DSPPGMADP	Display Programs that Adopt
GRTUSRAUT	Grant User Authority (*from-user-profile*)
QSYCUSRS	Check User Special Authorities API
QSYLOBJA	List Authorized Objects API
QSYLOBJP	List Objects That Adopt API
QSYRUSRI	Retrieve User Information API
RTVUSRPRF	Retrieve User Profile
WRKOBJOWN	Work with Owned Objects
WRKUSRPRF	Work with User Profiles

Operations for User Queue (*USRQ):

- No Read or Change operations are audited for the *USRQ object type.

Operations for User Space (*USRSPC):

- Read operation

QUSRTVUS	Retrieve User Space API

- Change operation

QUSCHGUS	Change User Space API
QUSCUSAT	Change User Space Attributes API

- Operations that are not audited

Access Direct access to user space using MI instructions (only allowed for user domain queues in libraries specified in the QALWUSRDMN system value.

QUSRUSAT Retrieve User Space Attributes API

Operations for Workstation Customizing Object (*WSCST):

- Read operation

  Vary When a customized device is varied on

  RTVWSCST Retrieve Workstation Customizing Object Source (only when *TRANSFORM is specified for the device type)

  STRPRTWTR Start Printer Writer (only for spooled files that are printed to a customized printer using the host print transform function)

  Print When output is printed directly (not spooled) to a customized printer using the host print transform function

- Change operation

  None

- Operations that are not audited

  None

Bibliography

You may need to refer to other IBM manuals for more specific information about a particular topic. The following IBM AS/400 manuals contain information that you may need.

Backup and Recovery:

- *Advanced Backup and Recovery Guide*, SC41-8079, provides information about planning a backup and recovery strategy, the different types of media available to save and restore system data, as well as a description of how to record changes made to database files using journaling and how that information can be used for system recovery. This manual describes how to plan for and set up user auxiliary storage pools (ASPs), mirrored protection, and checksums along with other availability recovery topics. It also describes how to install the system again from backup.

  Short title: *Advanced Backup and Recovery Guide*

- *Basic Backup and Recovery Guide*, SC41-0036, contains a subset of the information found in the *Advanced Backup and Recovery Guide*, SC41-8079. The manual contains information about planning a backup and recovery strategy, the different types of media available to save and restore system data, save and restore procedures, and disk recovery procedures. It also describes how to install the system again from backup.

  Short title: *Basic Backup and Recovery Guide*

Basic Security Information and Physical Security:

- *Basic Security Guide*, SC41-0047, explains why security is necessary, defines major concepts, and provides information on planning, implementing, and monitoring basic security on the AS/400 system.

  Short title: *Basic Security Guide*

Communications and Networking:

- *Communications: Distribution Services Network Guide*, SC41-9588, provides information about configuring a network for Systems Network Architecture distribution services (SNADS) and the Virtual Machine/Multiple Virtual Storage (VM/MVS) bridge. In addition, object distribution functions, document library services, and system distribution directory services are discussed.

  Short Title: *Distribution Services Network Guide*

- *Communications: Remote Work Station Guide*, SC41-0002, provides information on how to set up and use remote work station support, such as display station pass-through, distributed host command facility, and 3270 remote attachment.

  Short Title: *Remote Work Station Guide*

- *Distributed Data Management Guide*, SC41-9600, provides information about remote file processing. It describes how to define a remote file to OS/400 distributed data management (DDM), how to create a DDM file, what file utilities are supported through DDM, and the requirements of OS/400 DDM as related to other systems.

  Short Title: *DDM Guide*

- *Transmission Control Protocol/Internet Protocol Guide*, SC41-9875, provides information about how the AS/400 system carries out TCP/IP. This guide describes how to use and configure TCP/IP and the TCP/IP applications of FTP, SMTP, and TELNET. It also provides information about the relationship of TCP/IP to other AS/400 communications protocols and the OfficeVision/400 licensed program.

  Short Title: *TCP/IP Guide*

C2 Security:

- *Guide to Enabling C2 Security*, SC41-0103, describes how to customize your system to meet the requirements for C2 Security, as described in the *Department of Defense Trusted Computer Evaluation Criteria*.

  Short title: *Guide to Enabling C2 Security*

- *Trusted Computer Systems Evaluation Criteria*, DoD 5200.28.STD, describes the criteria for levels of trust for computer systems. The TCSEC is a publication of the United States government. Copies may be obtained from:

  Office of Standards and Products
  National Computer Security Center
  Fort Meade, Maryland 20755-6000 USA
  Attention: Chief, Computer Security Standards

  Short title: *TCSEC*

General System Operations:

- *New User's Guide*, SC41-8211, provides beginner information about how to sign on and off; send and receive messages; respond to keyboard error messages; use function keys; use display, command, and help information; and control and manage jobs.

  Short title: *New User's Guide*

- *System Introduction*, GC41-9766, provides information about the features and capabilities of the AS/400 system, as well as the characteristics of the system and various IBM licensed programs.

  Short title: *System Introduction*

- *System Operator's Guide*, SC41-8082, provides information about how to use the system unit control panel and console; send and receive messages; respond to error messages; start and stop the system; and do such system tasks as working with jobs, printing, backup and recovery, messages, tapes and diskettes, online education, program temporary fixes (PTFs), and problems. Also included are sections on setting up the AS/400 system and keeping it running smoothly.

Short title: *Operator's Guide*

- *System Operator's Quick Reference*, SX41-9573, provides a summary of how to do day-to-day tasks on the AS/400 system.

  Short title: *Operator's Quick Reference*

IBM-Supplied Program Installation and System Configuration:

- *Device Configuration Guide*, SC41-8106, provides information about how to do an initial configuration and how to change that configuration. It also contains conceptual information about device configuration.

  Short title: *Device Configuration Guide*

- *Licensed Programs and New Release Installation Guide*, SC41-9878, provides step-by-step procedures for initial install, installing licensed programs, program temporary fixes (PTFs), and secondary languages from IBM.

  Short title: *Licensed Programs and New Release Installation Guide*

Migration and System/36 Environment:

- *Migrating from System/36 Planning Guide*, GC41-9623, provides information to help migrate products and applications using the AS/400 System/36 Migration Aid (program 5727-MG1). It includes information for planning the details of migration and an overview of the functions on the System/36 to AS/400 Migration Aid.

  Short Title: *Migrating from System/36 Planning Guide*

- *Migrating from System/38 Planning Guide*, GC41-9624, provides information to help migrate products and applications using the AS/400 System/38 Migration Aid (program 5714-MG1). It includes information for planning the details of migration and an overview of the functions on the System/38 to AS/400 Migration Aid.

  Short Title: *Migrating from System/38 Planning Guide*

- *Programming: Concepts and Programmer's Guide for the System/36 Environment*, SC41-9663, provides information identifying the differences in the applications process in the System/36 environment on the AS/400 system. It helps the user understand the functional and operational differences (from a System/36 perspective) when processing in the System/36 environment on the AS/400 system. This includes an environment functional overview, considerations for migration, programming, communications, security, and coexistence.

  Short Title: *Concepts and Programmer's Guide for the System/36 Environment*

- *Programming: System Reference for the System/36 Environment*, SC41-9662, provides information about using System/36 procedure control expressions, procedures, operation control language (OCL) statements, control commands, and utilities on the AS/400 system.

  Short Title: *System Reference for the System/36 Environment*

National Language Support:

- *National Language Support Planning Guide*, GC41-9877, provides information required to understand and use the national language support function on the AS/400 system. This manual prepares the AS/400 user for planning and using the national language support (NLS) and the multilingual support of the AS/400 system. It also provides an explanation of the database management of multilingual data and application considerations for a multilingual system.

  Short Title: *National Language Support Planning Guide*

OfficeVision/400 Licensed Program:

- *Office Services Concepts and Programmer's Guide*, SC41-9758, provides information about writing applications that use OfficeVision/400 functions. This manual also includes an overview of directory services, document distribution services, document library services, document and folder save and restore and storage management, security services, word processing services, and information on finding new ways to integrate your applications with OfficeVision/400.

  Short Title: *Office Services Concepts and Programmer's Guide*

- *Systems Application Architecture* OfficeVision/400*: Managing OfficeVision/400*, SC41-9627, provides information on how to manage the day-to-day activities of OfficeVision/400. It also includes information on maintaining office enrollment and creating and managing office objects.

  Short Title: *Managing OfficeVision/400**

- *Systems Application Architecture* OfficeVision/400*: Planning For and Setting Up OfficeVision/400*, SC41-9626, provides information about planning for and setting up OfficeVision/400. The information includes planning for enrolling users, word processing, mail and calendar processing, using OfficeVision/400 with IBM personal computers, and using OfficeVision/400 in a communications network. The planning activities include filling out planning worksheets that are used to do the setup tasks.

  Short Title: *Planning For and Setting Up OfficeVision/400**

PC Support/400 Licensed Program:

- *PC Support/400: DOS and OS/2 Technical Reference*, SC41-8091, provides technical information about the PC Support programs for all versions of PC Support.

  Short Title: *PC Support/400 Technical Reference for DOS and OS/2*

Printing:

- *Guide to Programming for Printing*, SC41-8194, provides information on printing elements and concepts of the AS/400 system, printer file and print spooling support for printing operation, and printer connectivity.

Short title: *Guide to Programming for Printing*

Programming:

- *Programming: Control Language Programmer's Guide*, SC41-8077, provides a wide-ranging discussion of AS/400 programming topics, including a general discussion of objects and libraries, CL programming, controlling flow and communicating between programs, working with objects in CL programs, and creating CL programs. Other topics include predefined and impromptu messages and message handling, defining and creating user-defined commands and menus, application testing, including debug mode, breakpoints, traces, and display functions.

  Short title: *CL Programmer's Guide*

- *Programming: Control Language Reference*, SC41-0030, provides a description of all the AS/400 control language (CL) and its OS/400 commands. The OS/400 commands are used to request functions of the Operating System/400 (5738-SS1) licensed program. All the non-OS/400 CL commands—those associated with the other AS/400 licensed programs, including all the various languages and utilities—are described in other manuals that support those licensed programs.

  Short title: *CL Reference*

- *Programming: Reference Summary*, SX41-0028, provides quick and easy access to summary information about many of the languages and utilities available on the AS/400 system. It contains summaries of:

  - All AS/400 CL commands (in OS/400 program and in all other licensed programs), in various forms.
  - Information related to CL commands, such as the error messages that can be monitored by each command, and the IBM-supplied files that are used by some commands.
  - IBM-supplied objects, including libraries.
  - IBM-supplied system values.
  - DDS keywords for physical, logical, display, printer, and ICF files.
  - REXX instructions and built-in functions.
  - Other languages (like RPG) and utilities (like SEU and SDA).

  Short title: *Programming Reference Summary*

- *Programming: Work Management Guide*, SC41-8078, provides information about how to create and change a work management environment. Other topics include a description of tuning the system, collecting performance data including information on record formats and contents of the data being collected, working with system values to control or change the overall operation of the system, and a description of how to gather data to determine who is using the system and what resources are being used.

  Short title: *Work Management Guide*

- *System Programmer's Interface Reference*, SC41-8223, provides information on how to create, use, and delete objects that help manage system performance, use spooling efficiently, and maintain database files efficiently. This manual also includes information on creating and maintaining the programs for system objects and retrieving OS/400 information by working with objects, database files, jobs, and spooling.

  Short Title: *System Programmer's Interface Reference*

Utilities:

- *Application Development Tools: Programming Development Manager User's Guide and Reference*, SC09-1339, provides information about using the Application Development Tools programming development manager (PDM) to work with lists of libraries, objects, members, and user-defined options to easily do such operations as copy, delete, and rename.

  This manual contains activities and reference material to help the user learn PDM. The most commonly used operations and function keys are explained in detail using examples.

  Short title: *PDM User's Guide and Reference*

- *Application Development Tools: Source Entry Utility User's Guide and Reference*, SC09-1338, provides information about using the Application Development Tools source entry utility (SEU) to create and edit source members. The manual explains how to start and end an SEU session and how to use the many features of this full-screen text editor. The manual contains examples to help both new and experienced users accomplish various editing tasks, from the simplest line commands to using pre-defined prompts for high-level languages and data formats.

  Short Title: *SEU User's Guide and Reference*

- *Query/400 User's Guide*, SC41-9614, describes how to use AS/400 Query to get data from any database file. It describes how to sign on to Query, and how to define and run queries to create reports containing the selected data. This manual describes all the tasks for defining the query of one or more files, the way the report is to look when the query is run, and whether the report is to be displayed, printed, or put in a database file.

  Short Title: *Query/400 User's Guide*

- *Systems Application Architecture* Structured Query Language/400 Programmer's Guide*, SC41-9609, provides an overview of how to design, write, run, and test SQL/400 statements. It also describes interactive Structured Query Language (SQL). The manual provides examples of how to write SQL statements in COBOL, RPG, C, FORTRAN, and PL/I programs.

  Short Title: *SQL/400* Programmer's Guide*

- *Systems Application Architecture* Structured Query Language/400 Query Manager User's Guide*, SC41-0037, provides information on how to:

  - Build, maintain, and run SQL queries
  - Create reports ranging from simple to complex

- Build, update, manage, query, and report on database tables using a forms-based interface
- Define and prototype SQL queries and reports for inclusion in application programs

Short Title: *SQL/400* Query Manager User's Guide*

Index

Special Characters

Customer Satisfaction Feedback

Application System/400
Security Reference
Version 2

Publication No. SC41-8083-02

Overall, how would you rate this manual?

	Very Satisfied	Satisfied	Dissatis-fied	Very Dissatis-fied
Overall satisfaction				

How satisfied are you that the information in this manual is:

Accurate				
Complete				
Easy to find				
Easy to understand				
Well organized				
Applicable to your tasks				
THANK YOU!				

Please tell us how we can improve this manual:

____

____

____

____

____

May we contact you to discuss your responses? __ Yes __ No
Phone: (____) __________ Fax: (____) __________

To return this form:

- Mail it
- Fax it
  United States and Canada: **800+937-3430**
  Other countries: **(+1)+507+253-5192**
- Hand it to your IBM representative.

Note that IBM may use or distribute the responses to this form without obligation.

____________________ ____________________
Name Address

____________________ ____________________
Company or Organization

____________________ ____________________
Phone No.

Customer Satisfaction Feedback
SC41-8083-02

IBM®

Cı
Al

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN DEPT 245
IBM CORPORATION
3605 HWY 52 N
ROCHESTER MN 55901-7899

Fold and Tape | **Please do not staple** | Fold and Tape

Cu
Al

Customer Satisfaction Feedback

Application System/400
Security Reference
Version 2

Publication No. SC41-8083-02

Overall, how would you rate this manual?

	Very Satisfied	Satisfied	Dissatisfied	Very Dissatisfied
Overall satisfaction				

How satisfied are you that the information in this manual is:

Accurate				
Complete				
Easy to find				
Easy to understand				
Well organized				
Applicable to your tasks				
THANK YOU!				

Please tell us how we can improve this manual:

____

____

____

____

____

May we contact you to discuss your responses? __ Yes __ No

Phone: (____) ___________ Fax: (____) ___________

To return this form:

- Mail it
- Fax it
  United States and Canada: **800+937-3430**
  Other countries: **(+1)+507+253-5192**
- Hand it to your IBM representative.

Note that IBM may use or distribute the responses to this form without obligation.

Name ____________________ Address ____________________

Company or Organization ____________________ ____________________

Phone No. ____________________ ____________________

Customer Satisfaction Feedback
SC41-8083-02

C
Al

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN DEPT 245
IBM CORPORATION
3605 HWY 52 N
ROCHESTER MN 55901-7899

Fold and Tape **Please do not staple** Fold and Tape

Cu
Al

Customer Satisfaction Feedback

Application System/400
Security Reference
Version 2

Publication No. SC41-8083-02

Overall, how would you rate this manual?

	Very Satisfied	Satisfied	Dissatisfied	Very Dissatisfied
Overall satisfaction				

How satisfied are you that the information in this manual is:

Accurate				
Complete				
Easy to find				
Easy to understand				
Well organized				
Applicable to your tasks				
THANK YOU!				

Please tell us how we can improve this manual:

____

____

____

____

____

May we contact you to discuss your responses? __ Yes __ No

Phone: (____) ___________ Fax: (____) ___________

To return this form:

- Mail it
- Fax it
  United States and Canada: **800+937-3430**
  Other countries: **(+1)+507+253-5192**
- Hand it to your IBM representative.

Note that IBM may use or distribute the responses to this form without obligation.

______________________________ ______________________________
Name Address

______________________________ ______________________________
Company or Organization

______________________________ ______________________________
Phone No.

Customer Satisfaction Feedback
SC41-8083-02

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN DEPT 245
IBM CORPORATION
3605 HWY 52 N
ROCHESTER MN 55901-7899

Fold and Tape **Please do not staple** Fold and Tape

C
A

Cu
Alc

Customer Satisfaction Feedback

Application System/400
Security Reference
Version 2

Publication No. SC41-8083-02

Overall, how would you rate this manual?

	Very Satisfied	Satisfied	Dissatis-fied	Very Dissatis-fied
Overall satisfaction				

How satisfied are you that the information in this manual is:

Accurate				
Complete				
Easy to find				
Easy to understand				
Well organized				
Applicable to your tasks				
THANK YOU!				

Please tell us how we can improve this manual:

____

____

____

____

____

May we contact you to discuss your responses? __ Yes __ No

Phone: (____) ___________ Fax: (____) ___________

To return this form:

- Mail it
- Fax it
  United States and Canada: **800+937-3430**
  Other countries: **(+1)+507+253-5192**
- Hand it to your IBM representative.

Note that IBM may use or distribute the responses to this form without obligation.

____________________ ____________________
Name Address

____________________ ____________________
Company or Organization

____________________ ____________________
Phone No.

Customer Satisfaction Feedback
SC41-8083-02

Cu
Al

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN DEPT 245
IBM CORPORATION
3605 HWY 52 N
ROCHESTER MN 55901-7899

Fold and Tape **Please do not staple** Fold and Tape

C
A

Program Number: 5738-SS1

Printed in Ireland by Printech International plc

SC41-8083-02