IBM

eserver

iSeries

Security Reference

Version 5

SC41-5302-06

IBM

@server

iSeries

Security Reference

Version 5

SC41-5302-06

Note

Before using this information and the product it supports, be sure to read the information in Appendix H, "Notices" on page 611.

Seventh Edition (September 2002)

This edition replaces SC41–5302–06. This edition applies only to reduced instruction set computer (RISC) systems.

Contents

Figures

Tables

About Security - Reference (SC41-5302)

This book provides information about planning, setting up, managing, and auditing security on your iSeries 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.

This book 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 iSeries Information Center (see "Prerequisite and related information" on page xvi) and the *Tips and Tools for Securing Your iSeries*, SC41-5300-06 book. Information on planning and setting up Basic System Security and Planning can also be found in the Information Center (see "Prerequisite and related information" on page xvi).

This book does not provide complete information about planning for IBM Lotus® Domino™ users. For Lotus Domino users, see the URL http://notes.net/notesua.nsf. This Web site provides information on IBM Lotus Notes™, Lotus Domino, and IBM Lotus Domino for iSeries. From this web site, you can download information in Domino database (.NSF) and Adobe Acrobat (.PDF) format, search databases, and find out how to obtain printed manuals.

This book does not contain complete information about the application programming interfaces (APIs) that are available to access security information. APIs are described in *System API Programming*, SC41-5800-00. This book does not contain information about the Internet. For information about considerations when you connect your system to the Internet see the IBM® SecureWay®: iSeries and the Internet in the Information Center (see "Prerequisite and related information" on page xvi). .

For a list of related publications, see the "Related information" on page 615.

Who should read this book

The primary audience for this book is the security administrator.

Chapter 9, "Auditing Security on the iSeries System" on page 247 is intended for anyone who wants to perform a security audit of the system.

This book 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.
- Use a query tool, such as the Query/400 licensed program.

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

Chapter 5, "Resource Security" on page 119

Chapter 6, "Work Management Security" on page 185

Chapter 7, "Designing Security" on page 207

Chapter 8, "Backup and Recovery of Security Information" on page 235

Conventions and terminology used in this book

The iSeries displays in this book could be shown as they are presented through iSeries Navigator, which is part of iSeries Access for Windows™ on the personal computer. The example displays in this book could also be shown without iSeries Navigator available.

For more information on using iSeries Navigator, refer to the iSeries Information Center (see"Prerequisite and related information").

Prerequisite and related information

Use the iSeries Information Center as a starting point for your iSeries information needs. It is available in either of the following ways:

- The Internet at this uniform resource locator (URL) address:

  `http://www.ibm.com/eserver/iseries/infocenter`
- On CD-ROM: SK3T–4090–00, iSeries Information Center. This package also includes the PDF versions of iSeries manuals (SK3T–4092–00, iSeries Information Center: Supplemental Manuals), which replaces the Softcopy Library CD-ROM.

The iSeries Information Center contains advisors and important topics such as CL commands, system application programming interfaces (APIs), logical partitions, clustering, Java™, TCP/IP, Web serving, and secured networks. It also includes links to related IBM Redbooks and Internet links to other IBM Web sites such as the Technical Studio and the IBM home page.

With every new hardware order, you receive the following CD-ROM information:

- **SK3T-4096-00, iSeries Installation and Service Library.** This CD-ROM contains PDF manuals needed for installation and system maintenance of an IBM @server iSeries.
- *iSeries Setup and Operations CD-ROM*, SK3T-4098-01. This CD-ROM contains IBM iSeries Access for Windows and the EZ-Setup wizard. iSeries Access Express offers a powerful set of client and server capabilities for connecting PCs to iSeries servers. The EZ-Setup wizard automates many of the iSeries setup tasks.

For a list of related publications, see the "Related information" on page 615.

iSeries Navigator

IBM iSeries Navigator is a powerful graphical interface for managing your iSeries systems that is a component of iSeries Access for Windows.. iSeries Navigator functionality includes system navigation, configuration, planning capabilities, and online help to guide you through your tasks. iSeries Navigator makes operation and administration of the server easier and more productive and is the only user interface to the new, advanced features of the OS/400 operating system. It also includes Management Central for managing multiple servers from a central server.

For more information on iSeries Navigator, see the Information Center.

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other iSeries documentation, fill out the readers' comment form at the back of this book.

- If you prefer to send comments by mail, use the readers' comment form with the address that is printed on the back. If you are mailing a readers' comment form from somewhere other than the United States, you can give the form to the local IBM branch office or IBM representative for postage-paid mailing.
- If you prefer to send comments by FAX, use either of the following numbers:
  - United States and Canada: 1-800-937-3430
  - Other countries: 1-507-253-5192
- If you prefer to send comments electronically, use this network ID:
  - IBMMAIL, to IBMMAIL(USIB56RZ)
  - RCHCLERK@us.ibm.com

Be sure to include the following:

- The name of the book.
- The publication number of the book.
- The page number or topic to which your comment applies.

What's new for V5R2

System service tools (SST)

There are several enhancements and additions to service tools for this release that make them easier to use and understand. You can now manage and create service tools user IDs from system service tools (SST) by selecting option 8 (Work with service tools user IDs) from the main SST display. You no longer need to go into dedicated service tools (DST) to reset passwords, grant or revoke privileges, or create service tools user IDs. **Note**: Information regarding Service tools has been moved to the Information Center. The following enhancements have been made to System service tools (SST):

- **Password management enhancements**

  The server is shipped with limited ability to change default and expired passwords. This means that you cannot change service tools user IDs that have default and expired passwords through the Change Service Tools User ID (QSYCHGDS) API, nor can you change their passwords through SST. You can only change a service tools user ID with a default and expired password through DST. And, you can change the setting to allow default and expired passwords to be changed. Also, you can use the new Start service tools (STRSST) privilege to create a service tools user ID that can access DST, but can be restricted from accessing SST.

- **Terminology changes**

  The textual data and other documentation have been changed to reflect the new service tools terminology. Specifically, the term service tools user IDs replaces previous terms, such as DST user profiles, DST user IDs, service tools user profiles, or variations of these names.

For information on how to work with Service tools, see the Information Center (see "Prerequisite and related information" on page xvi for details) topic, Service tools (**Security—>Service tools**).

Audit file enhancements

There is a new outfile format, TYPE5, for the security audit journal. All new fields in existing records will only be added to the TYPE5 outfiles, and new audit records will only have TYPE5 outfiles. When you display the security audit journal, you can specify *TYPE5 as the outfile format for the journal. TYPE5 outfiles include remote IP information and thread IDs in the header portion of the security audit records. This data was added to the header portion of the audit record to keep the size of the records to a minimum. The remote IP information is helpful with intrusion detection and in determining the source of some actions on your system. The thread ID is useful for isolating actions within a thread in multi-threaded jobs. You can find more information about audit journals in Appendix F: in Security Reference.

Independent Disk Pool (IASP) security considerations

If you are using IASPs you need to understand the security implications of using them. You can find more information about how security affects and is affected by IASPs in Chapter 1 of Security Reference.

New values for QFRCCVNRST system value

You can now specify a wider range of values for the force conversion on restore (QFRCCVNRST) system value. The values that you specify for this and two other system values affect how your system handles restore operations. These three system values work together to provide options for controlling how trusted an object must be before it is restored on your system. You can find more information about the security implications of these system values in Chapter 3 of Security Reference.

Allow change of security related system values

System service tools (SST) and dedicated service tools (DST) provide an option that allows you to prevent changes to a variety of security related system values. If the value of the Allow change of security related system values option is set to NO, then the system values cannot be changed by using the Change system value (CHGSYSVAL) command (or any other user interfaces). Setting this option to NO is useful, for example, if you have settings for the Verify objects during restore (QVFYOBJRST) or Allow restore of security-sensitive objects (QALWOBJRST) system values to control how trusted an object must be before it can be restored. Selecting NO for this option ensures that applications cannot change these system value settings during install to values that are less restrictive to install objects that do not satisfy the settings for these system values. You can find information about restriction certain system values from being changed in Chapter 3 of Security Reference.

Finding "commonly-used" topics:

- "Defining How Information Can Be Accessed" on page 120
- "How the System Checks Authority" on page 156
- "Output Queue and Parameter Authorities Required for Printing" on page 199
- "Planning the Auditing of Object Access" on page 263
- Appendix D, "Authority Required for Objects Used by Commands" on page 309

Chapter 1. Introduction to iSeries Security

The @server 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 thousand 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 iSeries 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?

To facilitate installation, many of the security capabilities on your system are not activated when your system is shipped. Recommendations are provided in this book to bring your system to a reasonable level of security. Consider the security requirements of your own installation as you evaluate the recommendations.

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.

Note: You must order the keylock feature on some models.

Physical security is described in the Information Center (see "Prerequisite and related information" on page xvi for details).

Keylock Security

The keylock on the 940x control panel controls access to various system control panel functions. The keylock position can be retrieved and changed under program control by using either of the following:

- Retrieve IPL Attributes (QWCRIPLA) API
- Change IPL Attributes (CHGIPLA) command

This allows the remote user access to additional functions available at the control panel. For example, it controls where the machine will IPL from and to what environment, either OS/400® or Dedicated Service Tools (DST).

The OS/400 System Value, QRMTSRVATR, controls the remote access. This value is shipped defaulted to off which will not allow the keylock to be overridden. The system value can be changed to allow remote access, but does require *SECADM and *ALLOBJ special authorities to change.

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 five levels of security:

Level 10:
Level 10 is no longer supported. See Chapter 2, "Using System Security (QSecurity) System Value" on page 9 for information about security levels (10, 20, 30, 40, and 50).

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 iSeries systems with high security requirements, and it is designed to meet C2 security requirements.

The system security levels are described in Chapter 2, "Using System Security (QSecurity) System Value" on page 9.

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, "Security System Values" on page 23.

Signing

A key component of security is integrity: being able to trust that objects on the system have not been tampered with or altered. Your operating system software is protected by digital signatures, and now you can reinforce integrity by signing software objects which you rely on (for more information on using signing to protect your system, see *Tips and Tools for Securing Your iSeries*). This is particularly important if the object has been transmitted across the internet or stored on media which you feel might have been modified. The digital signature can be used to detect if the object has been altered.

Digital signatures, and their use for verification of software integrity, can be managed according to your security policies using the Verify Object Restore (QVFYOBJRST) system value, the Check Object Integrity (CHKOBJITG) command, and the Digital Certificate Manager tool. Additionally, you can choose to sign your own programs (all licensed programs shipped with the iSeries are signed). DCM is described in the Information Center (see "Prerequisite and related information" on page xvi for details).

New for V5R2, you can restrict adding digital signatures to a digital certificate store using the Add Verifier API and restrict resetting passwords on the digital certificate store. System Service Tools (SST) provides a new menu option, entitled "Work with system security" where you can restrict adding digital certificates.

Single sign-on enablement

In today's heterogeneous networks with partitioned servers and multiple platforms, administrators must cope with the complexities of managing identification and authentication for network users. IBM's new infrastructure and exploitation of it in iSeries helps administrators, users, and application programmers to much more cheaply and easily manage these identification and authentication.

To enable a single sign-on environment, IBM provides two technologies that work together to allow users to sign in with their Windows username and password and be authenticated to iSeries systems in the network. Network authentication service and Enterprise Identity Mapping (EIM) are the two technologies that an administrator must configure to enable a single sign-on environment. Windows 2000, XP, AIX, and zSeries use Kerberos protocol to authenticate users to the network. A secure, centralized server, called a key distribution center, authenticates principals (Kerberos users) to the network.

While network authentication service allows an iSeries system to participate in that Kerberos realm, EIM provides a mechanism for associating these Kerberos principals to a single EIM identifier that represents that user within the entire enterprise. Other user identities, such as an OS/400 username, can also be associated with this EIM identifier. When a user signs onto the network and accesses an iSeries system, he or she is not prompted for a userid and password. If the Kerberos authentication is successful, applications can look up the association to the EIM identifier to find the OS/400 username. The user no longer needs a password to iSeries applications and functions because the user is already authenticated through the Kerberos protocol. Administrators can centrally manage user identities with EIM while network users need only to manage one password. You can enable single sign-on by configuring network authentication service and Enterprise Identity Mapping (EIM) on your iSeries system. To review a scenario that shows how to set up a single sign-on environment, see the Information Center topic, Scenario: Enable single sign-on.(**Security—>Network authentication service—>Network authentication service scenarios—>Scenario: Enable single sign-on**. See "Prerequisite and related information" on page xvi for more information on accessing the Information Center.

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:

Special authority
: Special authorities 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, "User Profiles" on page 63.

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 229 discusses using group authority. "Group Ownership of Objects" on page 129 discusses what objects should be owned by group profiles. "Primary Group for an Object" on page 130 discusses using primary group and primary group authority for an object. "Copying User Profiles" on page 107 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 for 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.

Primary group
You can specify a primary group for an object. The primary group's authority is stored with the object. Using primary groups may simplify your authority management and improve authority checking performance.

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.

Directory authority
You can use directory authority in the same way that you use library authority. You can group objects in a directory and secure the directory rather than the individual objects.

Object authority
In cases where restricting access to a library or directory 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 information 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.

Field level authority
Field level authorities are given to individual fields in a database file. This authority is managed through SQL.

Resource security is described in Chapter 5, "Resource Security" on page 119

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, "Auditing Security on the iSeries System" on page 247 provides information about auditing security.

C2 Security

By using security level 50 and following the instructions in the *Security - Enabling for C2*, SC41-5303-00, you can bring a Version 4 Release 4 iSeries® 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).

In October, 1995, iSeries formally received a C2 security rating from the United States Department of Defense. The C2 rating is for V2R3 of OS/400, SEU, Query/400, SQL, and Common Cryptographic Architecture Services/400. The C2 rating was awarded after a rigorous, multi-year period of evaluation. iSeries is the first system to achieve a C2 rating for a system (hardware and operating system) with an integrated, full-function database.

In 1999, iSeries received a C2 rating for Version 4 Release 4 of OS/400 (with feature code 1920), SEU, Query/400, SQL, TCP/IP Utilities, Cryptographic Access Provider, and Advanced Series Hardware. A limited set of TCP/IP communication functions between iSeries, attached to a local area network, were included in the evaluation.

To achieve a C2 rating, a system must meet strict criteria in the following areas:

- Discretionary access control
- User accountability
- Security auditing

- Resource isolation

Independent disk pool

Independent disk pools provide the ability to group together storage that can be taken offline or brought online independent of system data or other unrelated data. The terms independent auxiliary storage pool (ASP) and independent disk pool are synonymous. An independent disk pool can be either switchable among multiple systems in a clustering environment or privately connected to a single system. For V5R2, functional changes to independent disk pools have security implications on your system. For example, when you perform a CRTUSRPRF, you can not create a user profile (*USRPRF) into an independent disk pool. However, when a user is privately authorized to an object in the independent disk pool, is the owner of an object on an independent disk pool, or is the primary group of an object on an independent disk pool, the name of the profile is stored on the independent disk pool. If the independent disk pool is moved to another system, the private authority, object ownership, and primary group entries will be attached to the profile with the same name on the target system. If a profile does not exist on the target system, a profile will be created. The user will not have any special authorities and the password will be set to *NONE.

Independent disk pools have been enhanced to provide support for library-based objects. In previous releases, independent disk pools supported user-defined file systems (UDFS) only. However several objects are not allowed on independent disk pools. For a complete list of supported and unsupported objects see Supported and unsupported OS/400 object types topic in the Information Center. (**Systems management—>Independent disk pools—>Concepts—>Restrictions and considerations—>Supported and unsupported OS/400 object types**).

Chapter 2. Using System Security (QSecurity) System Value

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: Before changing on a production system, read appropriate section on migrating from one level to another.

The system offers five levels of security:

10 **No system-enforced security**

Note: You cannot set the system value QSECURITY to security level 10.

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.

Your system is shipped at level 40, which provides sign-on and resource security and provides integrity protection. For more information, see "Security Level 40" on page 14.

If you want to change the security level, use the Work with System Values (WRKSYSVAL) command. The minimum security level you should use is 30. However, level 40 or higher is recommended. The change takes effect the next time you perform an initial program load (IPL). Table 1 compares the levels of security on the system:

Table 1. Security Levels: Function Comparison

Function	Level 20	Level 30	Level 40	Level 50
User name required to sign on.	Yes	Yes	Yes	Yes
Password required to sign on.	Yes	Yes	Yes	Yes
Password security active.	Yes	Yes	Yes	Yes
Menu and initial program security active.	Yes[1]	Yes[1]	Yes[1]	Yes[1]
Limit capabilities support active.	Yes	Yes	Yes	Yes
Resource security active.	No	Yes	Yes	Yes
Access to all objects.	Yes	No	No	No
User profile created automatically.	No	No	No	No

Table 1. Security Levels: Function Comparison (continued)

Function	Level 20	Level 30	Level 40	Level 50
Security auditing capabilities available.	Yes	Yes	Yes	Yes
Programs that contain restricted instructions cannot be created or recompiled.	Yes	Yes	Yes	Yes
Programs that use unsupported interfaces fail at run time.	No	No	Yes	Yes
Enhanced hardware storage protection supported.	No	No	Yes	Yes
Library QTEMP is a temporary object.	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
Pointers used in parameters are validated for user domain programs running in system state.	No	No	Yes	Yes
Message handling rules are enforced between system and user state programs.	No	No	No	Yes
A program's associated space cannot be directly modified.	No	No	Yes	Yes
Internal control blocks are protected.	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 20.

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.

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

***IOSYSCFG**
System configuration special authority allows a user to configure input and output devices on the system.

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

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

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

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

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

New for V5R2, you can also restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 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. Default Special Authorities for User Classes by Security Level

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

Note: The topics “User Class” on page 69 and “Special Authority” on page 75 provide more information about user classes and special authorities.

Recommendations:

Security level 30 or higher 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.

Also, at security level 30 (or below), users are able to call system interfaces that swap to QSECOFR user profile or allow users access to resources that they would not normally be allowed to access. At security level 40, users are not allowed to directly call these interfaces; therefore, security level 40 or higher is strongly recommended.

Security level 40 provides additional integrity protection without affecting system performance. Applications that do not run at security level 40 have a negative affect on performance at security level 30. They cause the system to respond to domain violations.

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 no security protection; therefore, security level 10 is **not recommended** by IBM. Beginning in Version 4 Release 3, you cannot set your security level to 10. If your system is currently at level 10, your system will remain at level 10 when you install Version 4 Release 3. If you change the system level to some other value, you cannot change it back to level 10.

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 successfully 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:

- 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.

All profiles are created with *ALLOBJ special authority at security level 20 by default. Therefore, security level 20 is **not recommended** by IBM.

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 133 in Appendix B for any changes you want to make to the profiles automatically created at security level 10.

Changing to Level 20 from a Higher Level

When you change from a higher security level to level 20, special authorities are added to the user profiles. By doing this, the user has, at least, the default special authority for the user class. Refer to Table 2 on page 11 to see how special authorities differ between level 20 and higher security levels.

Attention: When you change to level 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 that the user was given at 10 or 20, but would not have at 30 or above, are removed. Special authorities that the user was given that are not associated with their user class are not changed. For example, *ALLOBJ special authority is removed from all user profiles except those with a user class of *SECOFR. See Table 2 on page 11 for a list of the default special authorities and the differences between level 10 or 20 and the 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 120 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 3 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 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 must be retranslated before it can be used.	No validation is performed. Program must be retranslated before it can be used.	No validation is performed. Program must be retranslated before it can be used.
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]
A user state program attempts to call or transfer control to a system domain program.	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.	Parameter validation is performed.	Parameter validation is performed.

Table 3. Comparison of Security Levels 30, 40, and 50 (continued)

Scenario Description	Level 30	Level 40	Level 50
An IBM*-supplied command is changed to run a different program using the CHGCMD command. The command is changed again to run the original IBM-supplied program, which is a system domain program. A user attempts to run the command.	Attempt is successful.	AF journal entry;[1,2,4] operation fails.[2,4]	AF journal entry;[1,2,4] operation fails.[2,4]

[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.

[4] When you change an IBM-supplied command, it can no longer call a system domain program.

If you use the auditing function at lower security levels, the system logs journal entries for most of the actions shown in Table 3 on page 14, 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.

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 or by *INHERIT state programs that are called 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 following commands:

  – Display Program (DSPPGM) to display the domain of a program
  – Display Service Program (DSPSRVPGM) to display the domain of a service program

- **State:**

  Programs are either *SYSTEM state, *INHERIT state, or *USER state. The *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 and *INHERIT states are reserved for IBM-supplied programs.

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

Table 4 shows the domain and state access rules:

Table 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.

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 a User ID and 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 “Subsystem Descriptions” on page 191 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 iSeries 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.

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 40 or 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.

Validation of Programs Being Restored

When a program is created, the iSeries 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 do not match, the actions taken by the system are controlled by the QFRCCVNRST and QALWOBJRST system values.

In addition to a validation value, a program may optionally have a digital signature that can be verified upon restore. Any system actions related to digital signatures are controlled by the QVFYOBJRST and QFRCCVNRST system values. The three system values, Verify Object on Restore (QVFYOBJRST), Force Conversion on Restore (QFRCCVNRST) and Allow Object Restore (QALWOBJRST),

act as a series of filters to determine whether a program will be restored without change, whether it will be re-created (converted) as it is restored, or whether it will not be restored to the system.

The first filter is QVFYOBJRST system value. It controls the restore operation on some objects that can be digitally signed. After an object is successfully checked and is validated by this system value, the object proceeds to the second filter, QFRCCVNRST system value. This system value allows you to specify whether or not to convert programs, service programs, or module objects during a restore operation. This system value also prevents certain objects from being restored. Only when the objects have passed the first two filters do they proceed to the final filter, QALWOBJRST system value. This system value controls whether or not objects with security sensitive attributes can be restored.

Programs created for the iSeries can contain information that allows the program to be re-created at restore time, without requiring the program source. Programs created for iSeries Version 5, Release 1 and later contain the information needed for re-creation even when the observability of the program is removed. Programs created for releases prior to Version 5, Release 1 can only be re-created at restore time if the observable information of the program has not been deleted.

Each of these system values are described in the Chapter 3, " Security System Values" in the section, entitled Security-Related Restore System Values.

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 267 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 17 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.

Attention: If you move from level 40 to level 20, some special authorities are added to all user profiles. (See Table 2 on page 11.) 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 book *Security - Enabling for C2*.

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)
- Restricting message handling between user and system state programs
- Preventing modification of all internal control blocks
- Making the QTEMP library a temporary object

Restricting User Domain Objects

Most objects are created in the 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:

- At security level 50, no job can get addressability to the QTEMP library for another job. Therefore, if user domain objects are stored in the QTEMP library, they cannot be used to pass information to another user.

  Because of the difference in handling the QTEMP library at security level 50, objects in the QTEMP library may not be deleted when you IPL after the system ends abnormally. You may need to run the Reclaim Storage (RCLSTG) command more often at security level 50. Objects that are in a user's QTEMP library when the system ends abnormally appear in the QRCL library and need to be deleted after running the RCLSTG command.
- 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 25 for more information.

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, cannot be modified by a user state program.

At security level 50, no system 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 18 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 19.
- Recompile any COBOL programs that assign the device in the SELECT clause to WORKSTATION if the COBOL programs were compiled using a pre-V2R3 compiler.
- Recompile any S/36 environment COBOL programs that were compiled using a pre-V2R3 compiler.
- Recompile any RPG/400* or System/38™ environment RPG* programs that use display files if they were compiled using a pre-V2R2 compiler.

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.

Attention: If you move from level 50 to level 20, some special authorities are added to all user profiles. This removes resource security protection. (See Table 2 on page 11.)

Chapter 3. Security System Values

This chapter describes the system values that control security on your system. System values allow you to customize many characteristics of your system. A group of system values are used to define system-wide security settings.

New for V5R2, you can restrict users from changing several security-related system values. These restrictions can prevent even a user with *SECADM and *ALLOBJ authority from changing these system values with the CHGSYSVAL command. In addition to restricting changes to these system values, you can also restrict adding digital cerificates to digital certificate store with the Add Verifier API and restrict password resetting on the digital certificate store.

The following system values can be restricted:

Table 5. System values that can be restricted

QALWOBJRST	QAUTOVRT	QLMTDEVSSN	QPWDLVL	QRMTSIGN
QALWUSRDMN	QCRTAUT	QLMTSECOFR	QPWDMAXLEN	QRMTSRVATR
QAUDCTL	QCRTOBJAUD	QMAXSGNACN	QPWDMINLEN	QSECURITY
QAUDENACN	QDEVRCYACN	QMAXSIGN	QPWDPOSDIF	QSHRMEMCTL
QAUDFRCLVL	QDSPSGNINF	QPWDEXPITV	QPWDRQDDGT	QUSEADPAUT
QAUDLVL	QDSCJOBITV	QPWDLMTAJC	QPWDRQDDIF	QVFYOBJRST
QAUTOCFG	QFRCCVNRST	QPWDLMTCHR	QPWDVLDPGM	
QAUTORMT	QINACTMSGQ	QPWDLMTREP	QRETSVRSEC	

You can use System Service Tools (SST) to enable these restrictions. To work with these restrictions, you must have a service tools user ID and password. Dedicated Service Tools also provides the ability to turn on and off these restrictions.

To enable these restrictions, complete the following:

1. From a command line, enter **STRSST**.
2. On the System Service Tools (SST) display, select Option 7, and press enter.
3. You can restrict security system values and new digital certificates by selecting **No** for each of these options.

The following sections discuss specific security system values. Those system values to which you can restrict changes are documented within their corresponding sections:

- General security system values
- Security-related system values
- Security-related restore system values
- System values that apply to 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) or password level (QPWDLVL 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

QFRCCVNRST
Force conversion on restore

QINACTITV
Inactive job time-out interval

QINACTMSGQ
Inactive job message queue

QLMTDEVSSN
Limit device sessions

QLMTSECOFR
Limit security officer

QMAXSIGN
Maximum sign-on attempts

QMAXSGNACN
Action when maximum sign-on attempts exceeded

QRETSVRSEC
Retain Server Security

QRMTSIGN
Remote sign-on requests

QSECURITY
Security level

QSHRMEMCTL
Shared memory control

QUSEADPAUT
Use Adopted Authority

QVFYOBJRST
Verify object on restore

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 *USRSPC, *USRIDX, *USRQ objects. The system cannot audit the movement of information to and from user domain objects.

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 6. Possible Values for the QALWUSRDMN System Value:

***ALL**	User domain objects are allowed in all libraries and directories on the system.
***DIR**	User domain objects are allowed in all directories 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: For most systems, the recommended value is *ALL. 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.

Some systems have application software that relies on object types *USRSPC, *USRIDX, or *USRQ. For those systems, the list of libraries for the QALWUSRDMN system value should include the libraries that are used by the application software. The public authority of any library placed in QALWUSRDMN, except QTEMP, should be set to *EXCLUDE. This limits the number of users that may use MI interface, that cannot be audited, to read or change the data in user domain objects in these libraries.

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.

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 7. Possible Values for the QCRTAUT System Value:

***CHANGE**	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

The QCRTAUT system value is not used for objects created in directories in the enhanced file system.

Attention: Several IBM-supplied libraries, including QSYS, have a CRTAUT value of *SYSVAL. If you change the QCRTAUT system value to something other than *CHANGE, you may encounter problems with signing on at new or automatically created devices. To avoid these problems when you change QCRTAUT to something other than *CHANGE, you should ensure that all device descriptions and their associated message queues have a PUBLIC authority of *CHANGE. One way to accomplish this is to change the CRTAUT value for library QSYS to *CHANGE from *SYSVAL.

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
```

New for V5R2, you can also restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 8. Possible Values for the QDSPSGNINF System Value:

0	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

Emulation sessions through iSeries Access are included. Local jobs that are signed on to a remote system are excluded. Jobs that are connected by file transfer protocol (FTP) are excluded. Prior to Version 4, Release 2, telnet jobs were also excluded. To control the time-out of FTP connections, change the INACTTIMO parameter on the Change FTP Attribute (CHGFTPA) command. To control the time-out of telnet sessions prior to V4R2, use the Change Telnet Attribute (CHGTELNA) command.

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

- 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 iSeries Access job may appear inactive to the system if the user is performing PC functions such as editing a document without interacting with the iSeries 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.

Table 9. Possible Values for the QINACTITV System Value:

***NONE:**	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: 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.

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 10. Possible Values for QINACTMSGQ System Value:

***ENDJOB**	Inactive jobs are ended. If the inactive job is a group job,[1] all jobs associated with the group are also ended. If the job is part of a secondary job,[1] 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[1] 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 38 for more information. **Attention:** 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. If you assign QINACTMSGQ as the user's message queue, all messages in the user's message queue are lost during each IPL.

[1] The *Work Management* book describes group jobs and secondary jobs.

Recommended Value: *DSCJOB unless your users run iSeries Access jobs. Using *DSCJOB when some iSeries Access 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 iSeries Access licensed program. The *CL Programming* book 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 iSeries Access 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.

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 11. 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. “Workstations” on page 187 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, QSRV, and QSRVBAS profiles, no matter how the QLMTSECOFR value is set.

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 12. 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 (that is, given *CHANGE authority) to the display station or if user profile QSECOFR is authorized (given *CHANGE authority) to the display station. This authority cannot come from public authority.
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 or because they have *ALLOBJ special 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.

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 13. Possible Values for the QMAXSIGN System Value:

3	A user can try to sign on a maximum of 3 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.

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 14. 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 system disables a device by varying it off. 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 system disables a user profile by changing the *Status* parameter to *DISABLED. 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 30 provides more information about the QSYSMSG message queue.

If the QSECOFR profile is disabled, you may sign on as QSECOFR at the console and enable the profile. If the console is varied off and no other user can vary it on, you must IPL the system to make the console available.

Recommended Value: 3.

Retain Server Security (QRETSVRSEC)

QRETSVRSEC system value determines whether decryptable authentication information associated with user profiles or validation list (*VLDL) entries can be retained on the host system. This does not include the iSeries user profile password.

If you change the value from 1 to 0, the system disables access to the authentication information. If you change the value back to 1, the system reenables access to the authentication information.

The authentication information can be removed from the system by setting the QRETSVRSEC system value to 0 and running the CLRSVRSEC (Clear Server Security Data) command. If you have a large number of user profiles or validation lists on your system the CLRSVRSEC command may run for an extensive period of time.

The encrypted data field of a validation list entry is typically used to store authentication information. Applications specify whether to store the encrypted data in a decryptable or non-decryptable form. If the applications choose a decryptable form and the QRETSVRSEC value is changed from 1 to 0, the encrypted data field information is not accessible from the entry. If the encrypted data field of a validation list entry is stored in a non-decryptable from, it is not affected by the QRETSVRSEC system value.

New for V5R2, you can also restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 15. Possible Values for the QRETSVRSEC System Value:

0	Server security data is not retained.
1	Server security data is retained.

Recommended Value: 0.

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, the workstation function of the iSeries Access licensed program, and TELNET access.

New for V5R2, you can also restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 16. 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 not 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 if 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. For TELNET access, there is no action for *REJECT.
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 iSeries Access access. If you do allow pass-through or iSeries Access access, use *FRCSIGNON or *SAMEPRF.

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

Share Memory Control (QSHRMEMCTL)

The QSHRMEMCTL system value defines which users are allowed to use shared memory or mapped memory that has write capability. To change this system value, users must have *ALLOBJ and *SECADM special authorities. A change to this system value takes effect immediately.

New for V5R2, you can also restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 17. Possible Values for the QSHRMEMCTL System Value:.

0	Users cannot use shared memory, or use mapped memory that has write capability. This value means that users cannot use shared-memory APIs (for example, shmat() — Shared Memory Attach API), and cannot use mapped memory objects that have write capability (for example, mmap() — Memory Map a File API provides this function). Use this value in environments with higher security requirements.
1	Users can use shared memory or mapped memory that has write capability. This value means that users can use shared-memory APIs (for example, shmat() — Shared Memory Attach API), and can use mapped memory objects that have write capability (for example, mmap() — Memory Map a File API provides this function).

Recommended Value: 1.

Use Adopted Authority (QUSEADPAUT)

The QUSEADPAUT system value defines which users can create programs with the use adopted authority (*USEADPAUT(*YES)) attribute. All users authorized by the QUSEADPAUT system value can create or change programs and service programs to use adopted authority if the user has the necessary authority to the program or service program.

The system value can contain the name of an authorization list. The user's authority is checked against this list. If the user has at least *USE authority to the named authorization list, the user can create, change, or update programs or service programs with the USEADPAUT(*YES) attribute. The authority to the authorization list cannot come from adopted authority.

If an authorization list is named in the system value and the authorization list is missing, the function being attempted will not complete. A message is sent indicating this.

However, if the program is created with the QPRCRTPG API, and the *NOADPAUT value is specified in the option template, the program creates successfully even if the authorization list does not exist.

If more than one function is requested on the command or API, and the authorization list is missing, the function is not performed. If the command being attempted when the authorization list cannot be found is Create Pascal Program (CRTPASPGM) or Create Basic Program (CRTBASPGM), the result is a function check.

New for V5R2, you can also restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 18. Possible Values for the QUSEADPAUT System Value:

authorization list name	A diagnostic message is signaled to indicate that the program is created with USEADPAUT(*NO) if all of the following are true: • An authorization list is specified for the QUSEADPAUT system value. • The user does not have authority to the authorization list mentioned above. • There are no other errors when the program or service program is created.
***NONE**	All users can create or change programs and service programs to use adopted authority if the users have the necessary authority to the program or service program.

Recommended Value: For production machines, create an authorization list with authority of *PUBLIC(*EXCLUDE). Specify this authorization list for the QUSEADPAUT system value. This prevents anyone from creating programs that use adopted authority.

You should carefully consider the security design of your application before creating the authorization list for QUSEADPAUT system value. This is especially important for application development environments.

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.

QAUTOCFG
Automatic device configuration

QAUTOVRT
Automatic configuration of virtual devices

QDEVRCYACN
Device recovery action

QDSCJOBITV [1]
Disconnected job time-out interval

QRMTSRVATR
Remote service attribute

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 Device Configuration (QAUTOCFG)

QAUTOCFG system value automatically configures locally attached devices. The value specifies whether devices that are added to the system are configured automatically.

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 19. Possible Values for the QAUTOCFG System Value:

0	Automatic configuration is off. You must configure manually any new local controllers or devices that you add to your system.
1	Automatic configuration is on. The system automatically configures any new local controllers or devices that you add to your system. The operator receives a message that indicates the changes to the system's configuration.

Recommended Value: When initiating system setup or when adding many new devices, the system value should be set to 1. At all other times the system value should be set at 0.

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 or telnet. Without automatic configuration, a user attempting to break in has a limited number of attempts at each virtual device. The limit is defined by the security officer using the QMAXSIGN system value. With automatic configuration active, the actual limit is higher. The system sign-on limit is multiplied by the number of virtual devices that can be created by the automatic configuration support. This support is defined by the QAUTOVRT system value.

1. This system value is also discussed in the Information Center (see "Prerequisite and related information" on page xvi for details).

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the "Work with system security" option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: "Security System Values".

Table 20. 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 Support* book has more information about using display station pass-through. The *TCP/IP Configuration and Reference* book as more information about using TELNET.

Device Recovery Action (QDEVRCYACN)

QDEVRCYACN specifies what action to take when an I/O error occurs for an interactive job's workstation.

New for V5R2, you can also restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the "Work with system security" option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: "Security System Values".

Table 21. Possible Values for the QDEVRCYACN System Value:

***DSCMSG**	Disconnects the job. When signing-on again, an error message is sent to the user's application program.
***MSG**	Signals the I/O error message to the user's application program. The application program performs error recovery.
***DSCENDRQS**	Disconnects the job. When signing-on again, a cancel request function is performed to return control of the job back to the last request level.
***ENDJOB**	Ends the job. A job log is produced for the job. A message indicating that the job ended because of the device error is sent to the job log and the QHST log. To minimize the performance impact of the ending job, the job's priority is lowered by 10, the time slice is set to 100 milliseconds and the purge attribute is set to yes.
***ENDJOBNOLIST**	Ends the job. A job log is not produced for the job. A message is sent to the QHST log indicating that the job ended because of the device error.

When a value of *MSG or *DSCMSG is specified, the device recovery action is not performed until the next I/O operation is performed by the job. In a LAN/WAN environment, this may allow one device to disconnect and another to connect,

using the same address, before the next I/O operation for the job occurs. The job may recover from the I/O error message and continue running to the second device. To avoid this, a device recovery action of *DSCENDRQS, *ENDJOB, or *ENDJOBNOLIST should be specified. These device recovery actions are performed immediately when an I/O error, such as a power-off operation, occurs.

Recommended Value:
*DSCMSG

Note: *ALLOBJ and *SECADM special authorities are not required to change this value.

Before Version 3, Release 6, the default value was *MSG. To leave as *MSG presents a potential security exposure.

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.

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 22. 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

Remote Service Attribute (QRMTSRVATR)

QRMTSRVATR controls the remote system service problem analysis ability. The value allows the system to be analyzed remotely. New for V5R2, you can also restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

The values allowed for the QRMTSRVATR system value are:

Table 23. Possible Values for the QRMTSRVATR System Value:

0	Remote service attribute is off.
1	Remote service attribute is on.

Recommended Value: 0

For information about remote access and the QRMTSRVATR system value, see "Keylock Security" on page 2.

Security-Related Restore System Values

Overview:

Purpose:
Controls how and which security-related objects are restored 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 not required.

Following are descriptions of system values that relate to restoring security-related objects on the system.

QVFYOBJRST
Verify object on restore

QFRCCVNRST
Force conversion on restore

QALWOBJRST
Allow restoring of security sensitive objects

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

Verify Object on Restore (QVFYOBJRST)

The QVFYOBJRST system value determines whether objects are required to have digital signatures in order to be restored to your system. You can prevent anyone from restoring an object, unless that object has a proper digital signature from a trusted software provider. This value applies to objects of types: *PGM, *SRVPGM, *SQLPKG, *CMD and *MODULE. It also applies to *STMF objects which contain Java programs.

When an attempt is made to restore an object onto the system, three system values work together as filters to determine if the object is allowed to be restored. The first filter is the verify object on restore QVFYOBJRST system value. It is used to control the restore of some objects that can be digitally signed. The second filter is the force conversion on restore QFRCCVNRST system value. This system value

allows you to specify whether or not to convert programs, service programs, SQL packages, and module objects during the restore. It can also prevent some objects from being restored. Only objects that can get past the first two filters are processed by the third filter. The third filter is the allow object on restore (QALWOBJRST) system value. It specifies whether or not objects with security-sensitive attributes can be restored.

If Digital Certificate Manager (OS/400 option 34) is not installed on the system, all objects except those signed by a system trusted source are treated as unsigned when determining the effects of the QVFYOBJRST system value during a restore operation.

A change to this system value takes effect immediately.

New for V5R2, you can also restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Attention

When your system is shipped, the QVFYOBJRST system value is set to 3. If you change the value of QVFYOBJRST, it is important to set the QVFYOBJRST value to 3 or lower before installing a new release of the OS/400 operating system.

Table 24. Possible Values for the QVFYOBJRST System Value:

1	Do not verify signatures on restore. Restore all objects regardless of their signature. This value should not be used unless you have signed objects to restore which will fail their signature verification for some acceptable reason.
2	Verify objects on restore. Restore unsigned commands and user-state objects. Restore signed commands and user-state objects, even if the signatures are not valid. This value should be used only if there are specific objects with signatures that are not valid which you want to restore. In general, it is dangerous to restore objects with signatures that are not valid on your system.

Table 24. Possible Values for the QVFYOBJRST System Value: (continued)

3	Verify signatures on restore. Restore unsigned commands and user-state objects. Restore signed commands and user-state objects only if the signatures are valid. This value may be used for normal operations, when you expect some of the objects you restore to be unsigned, but you want to ensure that all signed objects have signatures that are valid. Commands and programs you have created or purchased before digital signatures were available will be unsigned. This value allows those commands and programs to be restored. This is the default value.
4	Verify signatures on restore. Do not restore unsigned commands and user-state objects. Restore signed commands and user-state objects, even if the signatures are not valid. This value should be used only if there are specific objects with signatures that are not valid which you want to restore, but you do not want the possibility of unsigned objects being restored. In general, it is dangerous to restore objects with signatures that are not valid on your system.
5	Verify signatures on restore. Do not restore unsigned commands and user-state objects. Restore signed commands and user-state objects only if the signatures are valid. This value is the most restrictive value and should be used when the only objects you want to be restored are those which have been signed by trusted sources

Objects which have the system-state attribute and objects which have the inherit-state attribute are required to have valid signatures from a system trusted source. The only value which will allow a system-state or inherit-state object to restore without a valid signature is 1. Allowing such a command or program represents an integrity risk to your system. If you change the QVFYOBJRST system value to 1 to allow such an object to restore on your system, be sure to change the QVFYOBJRST system value back to its previous value after the object has been restored.

Some commands use a signature that does not cover all parts of the object. Some parts of the command are not signed while other parts are only signed when they contain a non-default value. This type of signature allows some changes to be made to the command without invalidating its signature. Examples of changes that will not invalidate these types of signatures include:

- Changing command defaults.
- Adding a validity checking program to a command that does not have one.
- Changing the 'where allowed to run' parameter.
- Changing the 'allow limited user' parameter.

If you wish, you can add your own signature to these commands that includes these areas of the command object.

Recommended Value: 3.

Force Conversion on Restore (QFRCCVNRST)

This system value allows you to specify whether or not to convert the following object types during a restore:

- program (*PGM)
- service program (*SRVPGM)
- SQL Package (*SQLPKG)
- module (*MODULE)

It can also prevent some objects from being restored. An object which is specified to be converted by the system value, but cannot be converted because it does not contain sufficient creation data, will not be restored.

The *SYSVAL value for the FRCOBJCVN parameter on the restore commands (RST, RSTLIB, RSTOBJ, RSTLICPGM) uses the value of this system value. Therefore, you can turn on and turn off conversion for the entire system by changing the QFRCCVNRST value. However, the FRCOBJCVN parameter overrides the system value in some cases. Specifying *YES and *ALL on the FRCOBJCVN will override all settings of the system value. Specifying *YES and *RQD on the FRCOBJCVN parameter is the same as specifying '2' for this system value and can override the system value when it is set to '0' or '1'.

QFRCCVNRST is the second of three system values that work consecutively as filters to determine if an object is allowed to be restored, or if it is converted during the restore. The first filter, verify object on restore (QVFYOBJRST) system value, controls the restore of some objects that can be digitally signed. Only objects that can get past the first two filters are processed by the third filter, the allow object restore (QALWOBJRST) system value, which specifies whether or not objects with security-sensitive attributes can be restored.

The shipped value of QFRCCVNRST is 1. For all values of QFRCCVNRST an object which should be converted but cannot be converted will not be restored. Objects digitally signed by a system trusted source are restored without conversion for all values of this system value.

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the with the"Work with system security" option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: "Security System Values".

The table below summarizes the allowed values for QFRCCVNRST:

Table 25. QFRCCVNRST Values

0	Do not convert anything. Do not prevent anything from being restored.
1	Objects with validation errors will be converted.
2	Objects will be converted if their conversion is required for the current operating system or if they have a validation error.
3	Objects which are suspected of having been tampered with, objects which contain validation errors, and objects which require conversion to be used on the current version of the operating system will be converted.
4	Objects which contain sufficient creation data to be converted and do not have valid digital signatures will be converted. An object that does not contain sufficient creation data will be restored without conversion. NOTE: Objects (signed and unsigned) which have validation errors are suspected of having been tampered with, or require conversion to be used on the current version of the operating system will be converted, or will fail to restore if they do not convert.
5	Objects that contain sufficient creation data will be converted. An object that does not contain sufficient creation data to be converted will be restored. NOTE: Objects which have validation errors, are suspected of having been tampered with, or require conversion to be used on the current version of the operating system that cannot be converted will not restore.
6	All objects which do not have a valid digital signature will be converted. NOTE: An object with a valid digital signature that also has a validation error or is suspected of having been tampered with will be converted, or if it cannot be converted, it will not be restored.
7	Every object will be converted.

When an object is converted, its digital signature is discarded. The state of the converted object is user state. Converted objects will have a good validation value and are not suspected of having been tampered with.

Recommended Value:3 or higher.

Allow Restoring of Security-Sensitive Objects (QALWOBJRST)

The QALWOBJRST system value determines whether objects that are security-sensitive may be restored to your system. You can use it to prevent anyone from restoring a system state object or an object that adopts authority.

When an attempt is made to restore an object onto the system, three system values work together as filters to determine if the object is allowed to be restored, or if it is converted during the restore. The first filter is the verify object on restore QVFYOBJRST system value. It is used to control the restore of some objects that can be digitally signed. The second filter is the force conversion on restore QFRCCVNRST system value. This system value allows you to specify whether or not to convert programs, service programs, SQL packages, and module objects during the restore. It can also prevent some objects from being restored. Only objects that can get past the first two filters are processed by the third filter. The third filter is the allow object on restore (QALWOBJRST) system value. It specifies whether or not objects with security-sensitive attributes can be restored.

When your system is shipped, the QALWOBJRST system value is set to *ALL. This value is necessary to install your system successfully.

ATTENTION: It is important to set the QALWOBJRST value to *ALL before performing some system activities, such as:

- Installing a new release of the OS/400 licensed program.
- Installing new licensed programs.
- Recovering your system.

These activities may fail if the QALWOBJRST value is not *ALL. To ensure system security, return the QALWOBJRST value to your normal setting after completing the system activity.

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

You may specify multiple values for the QALWOBJRST system value, unless you specify *ALL or *NONE.

Table 26. Possible Values for the QALWOBJRST System Value:.

***ALL**	Any object may be restored to your system by a user with the proper authority.
***NONE**	Security-sensitive objects, such as system state programs or programs that adopt authority, may not be restored to the system.
***ALWSYSSTT**	System and inherit state objects may be restored to the system.
***ALWPGMADP**	Objects that adopt authority may be restored to the system.
***ALWPTF**	System and inherit state objects, objects that adopt authority, objects that have the S_ISUID(set-user-ID) attribute enabled, and objects that have S_ISGID (set-group-ID) attribute enabled can be restored to the system during PTF install.
***ALWSETUID**	Allow restore of files that have the S_ISUID (set-user-ID) attribute enabled.
***ALWSETGID**	Allow restore of files that have the S_ISGID (set-group-ID) attribute enabled.
***ALWVLDERR**	Allow restore of objects that do not pass the object validation tests. If the setting of QFRCCVNRST system value causes the object to be converted, its validation errors will have been corrected.

Recommended Value: The QALWOBJRST system value provides a method to protect your system from programs that may cause serious problems. For normal operations, consider setting this value to *NONE. Remember to change it to *ALL before performing the activities listed previously. If you regularly restore programs and applications to your system, you may need to set the QALWOBJRST system value to *ALWPGMADP.

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

QPWDEXPITV [2]
Expiration interval

QPWDLVL
Password level

QPWDMINLEN [2]
Minimum length

QPWDMAXLEN [2]
Maximum length

QPWDRQDDIF [2]
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.

2. These system values are also discussed in the Information Center (see"Prerequisite and related information" on page xvi for details).

If the Password Minimum Length (QPWDMINLEN) system value has a value other than 1 or the Password Maximum Length (QPWDMAXLEN) system value has a value other than 10 or you change any of the other password-control system values from the defaults, 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
```

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 27. 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.

Password Level (QPWDLVL)

The password level of the system can be set to allow for user profile passwords from 1-10 characters or to allow for user profile passwords from 1-128 characters.

The password level can be set to allow a 'passphrase' as the password value. The term 'passphrase' is sometimes used in the computer industry to describe a password value which can be very long and has few, if any, restrictions on the characters used in the password value. Blanks can be used between letters in a

passphrase, which allows you to have a password value that is a sentence or sentence fragment. The only restrictions on a passphrase are that it cannot start with an asterisk ('*') and trailing blanks will be removed. Before changing the password level of your system, please review the section "Planning Password Level Changes" on page 209.

New for V5R2, you can also restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the "Work with system security" option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: "Security System Values".

Table 28. Possible Values for the QPWDLVL System Value:.

0	The system supports user profile passwords with a length of 1-10 characters. The allowable characters are A-Z, 0-9 and characters $, @, # and underscore. QPWDLVL 0 should be used if your system communicates with other iSeries systems in a network and those systems are running with either a QPWDLVL value of 0 or an operating system release less than V5R1M0. QPWDLVL 0 should be used if your system communicates with any other system that limits the length of passwords from 1-10 characters. QPWDLVL 0 must be used if your system communicates with the Windows® 95/98/ME iSeries Client Support for Windows Network Neighborhood (NetServer) product and your system communicates with other systems using passwords from 1-10 characters. When the QPWDLVL value of the system is set to 0, the operating system will create the encrypted password for use at QPWDLVL 2 and 3. The password value that can be used at QPWDLVL 2 and 3 will be the same password as is being used at QPWDLVL 0 or 1.
1	QPWDLVL 1 is the equivalent support of QPWDLVL 0 with the following exception: iSeries NetServer passwords for Windows 95/98/ME clients will be removed from the system. If you use the client support for the iSeries NetServer product you cannot use QPWDLVL value 1. QPWDLVL 1 improves the security of the iSeries system by removing all iSeries NetServer passwords from the system.

Table 28. Possible Values for the QPWDLVL System Value: (continued).

2	The system supports user profile passwords from 1-128 characters. Upper and lower case characters are allowed. Passwords can consist of any character and the password will be case sensitive. QPWDLVL 2 is viewed as a compatibility level. This level allows for a move back to QPWDLVL 0 or 1 as long as the password created on QPWDLVL 2 or 3 meets the length and syntax requirements of a password valid on QPWDLVL 0 or 1. QPWDLVL 2 can be used if your system communicates with the Windows 95/98/ME iSeries Client Support for Windows Network Neighborhood (NetServer) product as long as your password is 1-14 characters in length. QPWDLVL 2 cannot be used if your system communicates with other iSeries systems in a network and those systems are running with either a QPWDLVL value of 0 or 1 or an operating system release less than V5R1M0. QPWDLVL 2 cannot be used if your system communicates with any other system that limits the length of passwords from 1-10 characters. No encrypted passwords are removed from the system when QPWDLVL is changed to 2.
3	The system supports user profile passwords from 1-128 characters. Upper and lower case characters are allowed. Passwords can consist of any character and the password will be case sensitive. QPWDLVL 3 cannot be used if your system communicates with other iSeries systems in a network and those systems are running with either a QPWDLVL value of 0 or 1 or an operating system release less than V5R1M0. QPWDLVL 3 cannot be used if your system communicates with any other system that limits the length of passwords from 1-10 characters. QPWDLVL 3 cannot be used if your system communicates with the Windows 95/98/ME iSeries Client Support for Windows Network Neighborhood (NetServer) product. All user profile passwords that are used at QPWDLVL 0 and 1 are removed from the system when QPWDLVL is 3. Changing from QPWDLVL 3 back to QPWDLVL 0 or 1 requires a change to QPWDLVL 2 before going to 0 or 1. QPWDLVL 2 allows for the creation of user profile passwords that can be used at QPWDLVL 0 or 1 as long as the length and syntax requirements for the password meet the QPWDLVL 0 or 1 rules.

Changing the password level of the system from 1-10 character passwords to 1-128 character passwords requires careful consideration. If your system communicates with other systems in a network, then all systems must be able to handle the longer passwords.

A change to this system value takes effect at the next IPL. To see the current and pending password level values, use the CL command DSPSECA (Display Security Attributes).

Minimum Length of Passwords (QPWDMINLEN)

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

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 29. Possible Values for the QPWDMINLEN System Value:

6	A minimum of six characters are required for passwords.
minimum-number-of-characters	Specify a value of 1 through 10 when the password level (QPWDLVL) system value is 0 or 1. Specify a value of 1 through 128 when the password level (QPWDLVL) system value is 2 or 3.

Recommended Value: 6, 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.

New for V5R2, you can also restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 30. Possible Values for the QPWDMAXLEN System Value:

8	A maximum of eight characters for a password are allowed.
maximum-number-of-characters	Specify a value of 1 through 10 when the password level (QPWDLVL) system value is 0 or 1. Specify a value of 1 through 128 when the password level (QPWDLVL) system value is 2 or 3.

Recommended Value: 8.

Required Difference in Passwords (QPWDRQDDIF)

The QPWDRQDDIF system value controls whether the password must be different from 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.

Note: The value of the QPWDRQDDIF system value determines how many of these previous passwords are checked for a duplicate password.

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 31. Possible Values for the QPWDRQDDIF System Value:

Value	*Number of Previous Passwords Checked for Duplicates*
0	0 Duplicate passwords are allowed.
1	32
2	24
3	18
4	12
5	10
6	8
7	6
8	4

Recommended Value: Select a value of 5 or less to prevent the use of repeated passwords. Use a combination of the QPWDRQDDIF system value and the QPWDEXPITV (password expiration interval) system value to prevent a password from being reused for at least 6 months. For example, set the QPWDEXPITV system value to 30 (days) and the QPWDRQDDIF system value to 5 (10 unique passwords). This means a typical user, who changes passwords when warned by the system, will not repeat a password for approximately 9 months.

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.

The QPWDLMTCHR system value is not enforced when the password level (QPWDLVL) system value has a value of 2 or 3. The QPWDLMTCHR system value can be changed at QPWDLVL 2 or 3, but will not be enforced until QPWDLVL is changed to a value of 0 or 1.

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 32. 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.

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 33. 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 passwords that are easy to guess, such as the same character repeated several times.

When the password level (QPWDLVL) system value has a value of 2 or 3, the test for repeated characters is case sensitive. This means that a lowercase 'a' is not the same as an uppercase 'A'.

New for V5R2, you can also restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 34. 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.
2	The same character cannot be used consecutively in a password.

Table 35 on page 52 shows examples of what passwords are allowed based on the QPWDLMTREP system value.

Table 35. Passwords with Repeating Characters with QPWDLVL 0 or 1

Password Example	QPWDLMTREP Value of 0	QPWDLMTREP Value of 1	QPWDLMTREP Value of 2
A11111	Allowed	Not allowed	Not allowed
BOBBY	Allowed	Not allowed	Not allowed
AIRPLANE	Allowed	Not allowed	Allowed
N707UK	Allowed	Not allowed	Allowed

Table 36. Passwords with Repeating Characters with QPWDLVL 2 or 3

Password Example	QPWDLMTREP Value of 0	QPWDLMTREP Value of 1	QPWDLMTREP Value of 2
j222222	Allowed	Not allowed	Not allowed
ReallyFast	Allowed	Not allowed	Not allowed
Mom'sApPlePie	Allowed	Not allowed	Allowed
AaBbCcDdEe	Allowed	Allowed	Allowed

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.

When the password level (QPWDLVL) system value has a value of 2 or 3, the test for the same character is case sensitive. This means that a lowercase 'a' is not the same as an uppercase 'A'.

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the "Work with system security" option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: "Security System Values".

Table 37. 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.

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the "Work with system security" option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: "Security System Values".

Table 38. 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

Recommended Value: 1.

Password Approval Program (QPWDVLDPGM)

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

The topic "Using a Password Approval Program" discusses the requirements of the password approval program and shows an example.

A password approval program must reside in the system auxiliary storage pool (ASP) or a basic user ASP.

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the "Work with system security" option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: "Security System Values".

Table 39. Possible Values for the QPWDVLDPGM System Value:

***NONE**	No user-written program is used. This includes any password approval programs registered in the exit registration facility.
**REGFAC*	The validation program is retrieved from the registration facility, exit point QIBM_QSY_VLD_PASSWRD. More than one validation program can be specified in the registration facility. Each program will be called until one of them indicates that the password should be rejected or all of them have indicated the password is valid.
program-name	Specify the name of the user-written validation program, from 1 through 10 characters. A program name cannot be specified when the current or pending value of the password level (QPWDLVL) system value is 2 or 3.
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 *REGFAC or a program name is specified in the QPWDVLDPGM system value, one or more programs are called by the Change Password (CHGPWD) command or Change Password (QSYCHGPW) API. The programs are 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.

If a program name is specified in the QPWDVLDPGM system value, the system passes the following parameters to the password approval program:

Table 40. 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.
4 [1]	*CHAR	10	The name of the user.

1 Position 4 is optional.

If *REGFAC is specified in the QPWDVLDPGM system value, refer to the Security Exit Program information in the System API manual for information on the parameters passed to the validation program.

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.

Attention: 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 following control language (CL) program is an example of a password approval program when a program name is specified for QPWDVLDLVL. 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 &USER)
   DCL VAR(&NEW)       TYPE(*CHAR) LEN(10)
   DCL VAR(&OLD)       TYPE(*CHAR) LEN(10)
   DCL VAR(&RTNCD)     TYPE(*CHAR) LEN(1)
   DCL VAR(&USER)      TYPE(*CHAR) LEN(10)
   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  USRPRF(&USER)  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
```

The following control language (CL) program is an example of a password approval program when *REGFAC is specified for QPWDVLDLVL.

This example checks to make sure the new password is in CCSID 37 (or if it is in CCSID 13488 it converts the new password to CCSID 37), that the new password does not end in a numeric character, and that the new password does not contain the user profile name. The example assumes that a message file (PWDERRORS) has been created and message descriptions (PWD0001 and PWD0002) have been added to the message file. Additional calculations can be added to the program to check other criteria for passwords:

```
/*********************************************************/
/*                                                       */
/*  NAME:   PWDEXITPGM1 - Password validation exit 1     */
/*                                                       */
/*  Validates passwords when *REGFAC is specified for    */
/*  QPWDVLDPGM. Program is registered using the ADDEXITPGM*/
/*  CL command for the QIBM_QSY_VLD_PASSWRD exit point.  */
/*                                                       */
/*                                                       */
/*  ASSUMPTIONS: If CHGPWD command was used, password    */
/*  CCSID will be job default (assumed to be CCSID 37).  */
/*  If QSYCHGPW API was used, password CCSID will be     */
/*  UNICODE CCSID 13488.                                 */
/*********************************************************/

DCL &EXINPUT    *CHAR 1000
DCL &RTN        *CHAR 1

DCL &UNAME      *CHAR 10
DCL &NEWPW      *CHAR 256
DCL &NPOFF      *DEC 5 0
DCL &NPLEN      *DEC 5 0
DCL &INDX       *DEC 5 0
DCL &INDX       *DEC 5 0
DCL &INDX       *DEC 5 0
DCL &INDX       *DEC 5 0

DCL &XLTCHR2    *CHAR 2 VALUE(X'0000')
DCL &XLTCHR     *DEC 5 0
DCL &XLATEU     *CHAR 255 VALUE('................................ +
                                !"#$%&''()*+,-./0123456789:;<=>?+
                                @ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_+
                                `ABCDEFGHIJKLMNOPQRSTUVWXYZ{|}~.+
                                .................................+
```

```
                                    ................................+
                                    ................................+
                                    ................................')

DCL &XLATEC     *CHAR 255 VALUE('................................+
                                    ................................+
                                    ................................+
                                    ................................+
                                    .ABCDEFGHI.......JKLMNOPQR......+
                                    ..STUVWXYZ......................+
                                    ................................+
                                    ................................')

/*****************************************************************/
/* FORMAT OF EXINPUT IS:                                         */

/* POSITION   DESCRIPTION                                        */
/* 001 - 020  EXIT POINT NAME                                    */
/* 021 - 028  EXIT POINT FORMAT NAME                             */
/* 029 - 032  PASSWORD LEVEL (binary)                            */
/* 033 - 042  USER PROFILE NAME                                  */
/* 043 - 044  RESERVED                                           */
/* 045 - 048  OFFSET TO OLD PASSWORD (binary)                    */
/* 049 - 052  LENGTH OF OLD PASSWORD (binary)                    */
/* 053 - 056  CCSID OF OLD PASSWORD  (binary)                    */
/* 057 - 060  OFFSET TO NEW PASSWORD (binary)                    */
/* 061 - 064  LENGTH OF NEW PASSWORD (binary)                    */
/* 065 - 068  CCSID OF NEW PASSWORD  (binary)                    */
/* ??? - ???  OLD PASSWORD                                       */
/* ??? - ???  NEW PASSWORD                                       */
/*                                                               */
/*****************************************************************/


/*****************************************************************/
/*  Establish a generic monitor for the program.                 */
/*****************************************************************/

MONMSG     CPF0000
/* Assume new password is valid */
CHGVAR &RTN  VALUE('0') /* accept */
/* Get new password length, offset and value. Also get user name */
CHGVAR &NPLEN VALUE(%BIN(&EXINPUT 61 4))
CHGVAR &NPOFF VALUE(%BIN(&EXINPUT 57 4) + 1)
CHGVAR &UNAME VALUE(%SST(&EXINPUT 33 10))
CHGVAR &NEWPW VALUE(%SST(&EXINPUT &NPOFF &NPLEN))
/* If CCSID is 13488, probably used the QSYCHGPW API which converts */
/* the passwords to UNICODE CCSID 13488. So convert to CCSID 37, if */
/* possible, else give an error */
IF COND(%BIN(&EXINPUT 65 4) = 13488) THEN(DO)
    CHGVAR &INDX2 VALUE(1)
    CHGVAR &INDX3 VALUE(1)
   CVT1:
    CHGVAR &XLTCHR VALUE(%BIN(&NEWPW &INDX2 2))
    IF COND( (&XLTCHR *LT 1) *OR (&XLTCHR *GT 255) ) THEN(DO)
       CHGVAR &RTN  VALUE('3') /* reject */
       SNDPGMMSG MSG('INVALID CHARACTER IN NEW PASSWORD')
       GOTO DONE
    ENDDO
    CHGVAR %SST(&NEWPW &INDX3 1) VALUE(%SST(&XLATEU &XLTCHR 1))
    CHGVAR &INDX2  VALUE(&INDX2 + 2)
    CHGVAR &INDX3  VALUE(&INDX3 + 1)
    IF COND(&INDX2 > &NPLEN) THEN(GOTO ECVT1)
    GOTO CVT1
   ECVT1:
    CHGVAR &NPLEN VALUE(&INDX3 - 1)
    CHGVAR %SST(&EXINPUT 65 4) VALUE(X'00000025')
```

```
ENDDO

/* Check the CCSID of the new password value - must be 37     */
IF COND(%BIN(&EXINPUT 65 4) *NE 37) THEN(DO)
  CHGVAR &RTN  VALUE('3') /* reject */
  SNDPGMMSG MSG('CCSID OF NEW PASSWORD MUST BE 37')
  GOTO DONE
ENDDO

/* UPPERCASE NEW PASSWORD VALUE                       */
CHGVAR &INDX2 VALUE(1)
CHGVAR &INDX3 VALUE(1)
CVT4:
  CHGVAR %SST(&XLTCHR2 2 1) VALUE(%SST(&NEWPW &INDX2 1))
  CHGVAR &XLTCHR VALUE(%BIN(&XLTCHR2 1 2))
  IF COND( (&XLTCHR *LT 1) *OR (&XLTCHR *GT 255) ) THEN(DO)
    CHGVAR &RTN  VALUE('3') /* reject */
    SNDPGMMSG MSG('INVALID CHARACTER IN NEW PASSWORD')
    GOTO DONE
  ENDDO
  IF COND(%SST(&XLATEC &XLTCHR 1) *NE '.') +
  THEN(CHGVAR %SST(&NEWPW &INDX3 1) VALUE(%SST(&XLATEC &XLTCHR 1)))
  CHGVAR &INDX2  VALUE(&INDX2 + 1)
  CHGVAR &INDX3  VALUE(&INDX3 + 1)
  IF COND(&INDX2 > &NPLEN) THEN(GOTO ECVT4)
  GOTO CVT4
ECVT4:

/* CHECK IF LAST POSITION OF NEW PASSWORD IS NUMERIC */
IF COND(%SST(&NEWPW &NPLEN 1) = '0') THEN(GOTO ERROR1)
IF COND(%SST(&NEWPW &NPLEN 1) = '1') THEN(GOTO ERROR1)
IF COND(%SST(&NEWPW &NPLEN 1) = '2') THEN(GOTO ERROR1)
IF COND(%SST(&NEWPW &NPLEN 1) = '3') THEN(GOTO ERROR1)
IF COND(%SST(&NEWPW &NPLEN 1) = '4') THEN(GOTO ERROR1)
IF COND(%SST(&NEWPW &NPLEN 1) = '5') THEN(GOTO ERROR1)
IF COND(%SST(&NEWPW &NPLEN 1) = '6') THEN(GOTO ERROR1)
IF COND(%SST(&NEWPW &NPLEN 1) = '7') THEN(GOTO ERROR1)
IF COND(%SST(&NEWPW &NPLEN 1) = '8') THEN(GOTO ERROR1)
IF COND(%SST(&NEWPW &NPLEN 1) = '9') THEN(GOTO ERROR1)

/* CHECK IF PASSWORD CONTAINS USER PROFILE NAME        */
CHGVAR &UNLEN VALUE(1)
LOOP2:      /* FIND LENGTH OF USER NAME */
 IF COND(%SST(&UNAME &UNLEN 1) *NE ' ') THEN(DO)
   CHGVAR &UNLEN VALUE(&UNLEN + 1)
   IF COND(&UNLEN = 11) THEN(GOTO ELOOP2)
   GOTO LOOP2
 ENDDO
ELOOP2:
 CHGVAR &UNLEN VALUE(&UNLEN - 1)


/* CHECK FOR USER NAME IN NEW PASSWORD             */
IF COND(&UNLEN *GT &NPLEN) THEN(GOTO ELOOP3)
CHGVAR &INDX VALUE(1)
LOOP3:
  IF COND(%SST(&NEWPW &INDX &UNLEN) = %SST(&UNAME 1 &UNLEN))+
     THEN(GOTO ERROR2)
  IF COND((&INDX + &UNLEN + 1) *LT 128) THEN(DO)
     CHGVAR &INDX VALUE(&INDX + 1)
     GOTO LOOP3
  ENDDO
ELOOP3:

/* New Password is valid                        */
GOTO DONE
```

```
ERROR1:  /* NEW PASSWORD ENDS IN NUMERIC CHARACTER */
 CHGVAR &RTN  VALUE('3') /* reject */
 SNDPGMMSG TOPGMQ(*PRV) MSGTYPE(*ESCAPE) MSGID(PWD0001) MSGF(QSYS/PWDERRORS)
 GOTO DONE

ERROR2:  /* NEW PASSWORD CONTAINS USER NAME */
 CHGVAR &RTN  VALUE('3') /* reject */
 SNDPGMMSG TOPGMQ(*PRV) MSGTYPE(*ESCAPE) MSGID(PWD0002) MSGF(QSYS/PWDERRORS)
 GOTO DONE

DONE:
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

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

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

Table 41. Possible Values for the QAUDCTL System Value:

***NONE**	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, CHGDLOAUD, or CHGAUD 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.
***NOQTEMP**	Auditing is not performed for most actions if the object is in QTEMP library. See Chapter 9, “Auditing Security on the iSeries System” on page 247 for more details. You must specify this value with either *OBJAUD or *AUDLVL. See “Planning Security Auditing” on page 253 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.

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 42. Possible Values for the QAUDENDACN System Value:

***NOTIFY**	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.

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.

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 43. Possible Values for the QAUDFRCLVL System Value:

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

New for V5R2, you can restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 44. 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.
***NETCMN**	Violation detected by APPN® Filter support is logged.
***OBJMGT**	Object move and rename operations are logged.
***OFCSRV**	Changes to the system distribution directory and office mail actions are logged.
***OPTICAL**	Use of Optical Volumes is logged.
***PGMADP**	Obtaining authority from a program that adopts authority is logged.
***PGMFAIL**	System integrity violations are logged.
***PRTDTA**	Printing a spooled file, sending output directly to a printer, and sending output to a remote 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 253 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 library is *SYSVAL. The QCRTOBJAUD value is *CHANGE. If you 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.

New for V5R2, you can also restrict users with *SECADM and *ALLOBJ authorities from changing this security related system value with the CHGSYSVAL command. You can specify this restriction in the System Service Tools (SST) with the ″Work with system security″ option.

Note: This restriction applies to several other system values.
For details on how to restrict changes to security system values and a complete list of the affected system values, see Chapter 3: ″Security System Values″.

Table 45. 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 is 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 263 provides more information about methods for setting up object auditing on your system. You may also control the auditing value at the library level with the CRTOBJAUD parameter with the CRTLIB command and the CHGLIB command.

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:
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 for changes to users profiles

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 133 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 or special authorities for group profiles rather than for each individual user profile. A user may be a member of up to 16 group profiles.

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. The system also generates a group identification number (gid) for the profile. You can also designate a profile as a group profile at the time that you create it by specifying a value in the GID parameter. "Planning Group Profiles" on page 229 shows an example of setting up a group profile.

User-Profile Parameter Fields

User profiles can be created in the following ways:

- iSeries Navigator
- Management Central
- Character-based interface

When you create a user profile, the profile is given these authorities to itself: *OBJMGT, *CHANGE. These authorities are necessary for system functions and should not be removed.

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 the 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 Programming* book.

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. Some communications methods limit the user ID to eight characters. The Add User display also limits the user profile name to eight characters.
- 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.

One technique for assigning user profile names 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	HARRISBK
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:
128

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 when the QPWDLVL system value is set to 0 or 1. Passwords can be a maximum of 128 characters when the QPWDLVL system value is set to 2 or 3.

When the password level (QPWDLVL) system value is 0 or 1, the rules for specifying passwords are the same as those used for user profile names. When the first character of the password is a Q and the second character is a numeric character, the Q can be omitted on the Sign On display. 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. When QPWDLVL is 2 or 3, the

user must specify the password as Q12345 on the signon display if the user profile was created with a password of Q12345. An all numeric password is allowed when QPWDLVL is 2 or 3, but the user profile password must be created as all numeric.

When the password level (QPWDLVL) system value is 2 or 3, the password is case sensitive and can contain any character including blank characters. However, the password may not begin with an asterisk character ('*') and trailing blank characters are removed.

Note: Passwords can be created using double byte characters. However, a password containing double byte characters cannot be used to signon via the system signon screen. Passwords containing double byte characters can be created by the CRTUSRPRF and CHGUSRPRF commands and can be passed to the system APIs that support the password parameter.

One-way encryption is used to store the password on the system. 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 the password minimum length (QPWDMINLEN) system value is not 1 or the password maximum length (QPWDMAXLEN) system value is not 10 or any of the other password composition system values have been changed from the default values, a user cannot set the password equal to the user profile name using the CHGPWD command, the ASSIST menu, or the QSYCHGPW API.

See the topic "System Values That Apply to Passwords" on page 44 for information about setting the password composition system values.

Table 46. Possible Values for PASSWORD:

***USRPRF**	The password for this user is the same as the user profile name. When the password level (QPWDLVL) system value is 2 or 3, the password is the uppercased value of the user profile name. For profile JOHNDOE, the password would be JOHNDOE, not johndoe.
***NONE**	No password is assigned to this user profile. Sign-on is not allowed with this user profile. You can submit a batch job using a user profile with password *NONE if you have proper authority to the user profile.
user- password	A character string (128 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.
- 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 expired to *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 length of password and the characters that passwords can contain. If your system communicates with other systems, use the QPWDMAXLEN system value to limit the passwords length. At password levels 0 and 1, the QPWDLMTCHR system value can be used to specify characters that cannot be used in passwords.

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 password is changed. You can change the password by using either the CHGPWD or CHGUSRPRF command, or the QSYCHGPW API, 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 1). 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 1. Password Expiration Message

Table 47. 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 112 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)`.

Table 48. 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 49 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 49. Default Special Authorities by User Class

Special Authority	User Classes				
	***SECOFR**	***SECADM**	***PGMR**	***SYSOPR**	***USER**
*ALLOBJ	All	10 or 20	10 or 20	10 or 20	10 or 20
*SECADM	All	All			
*JOBCTL	All	10 or 20	10 or 20	All	
*SPLCTL	All				
*SAVSYS	All	10 or 20	10 or 20	All	10 or 20
*SERVICE	All				
*AUDIT	All				
*IOSYSCFG	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 50 on page 71 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 50. 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 98.

Table 51. Possible Values for ASTLVL:

***SYSVAL**	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 or *PARTIAL.

The topic "Library Lists" on page 193 provides more information about using library lists and the current library.

Table 52. Possible Values for CURLIB:

***CRTDFT**	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 *EXECUTE 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 or *PARTIAL, 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 185 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.

Table 53. 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.

Table 54. 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 *EXECUTE 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.

Table 55. Possible Values for MENU:

MAIN	The iSeries 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.

Table 56. 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 one 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)
Work with Messages (WRKMSG)

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, the Command Entry display or an option from a command grouping menu. Users are not restricted from doing the following:

- Running commands in CL programs that are running a command as a result of taking an option from a menu
- Running remote commands through applications.

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.

Possible Values: Table 57 shows the possible values for *Limit capabilities* and what functions are allowed for each value.

Table 57. 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	No	Yes
Change Attention Program	No	No	Yes
Enter Commands	A few[1]	Yes	Yes

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

Recommendations: Using an initial menu, restricting command line use, and providing access to the 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 217 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.

Table 58. 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 types of actions a user can perform on system resources. A user can be given one or more special authorities.

Table 59. 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 49 on page 70 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. A user with *SECADM special authority can:

- Add users to the system distribution directory.
- 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.

*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, change, end, hold, and release 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 "Printing" on page 197.

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 job queues, including holding, releasing, and clearing the job queue. 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.

*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 iSeries 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 start system service tools using the STRSST command. It also allows the user to debug a program with only *USE authority to the program and perform the display and alter service functions. The dump function can be performed without *SERVICE authority. It also allows the user to perform various trace functions.

Risks: A user with *SERVICE special authority can display and change confidential information using service functions. The user must have *ALLOBJ special authority to change the information using service functions.

To minimize the risk for trace commands, users can be given authorization to perform service tracing without needing to give the user *SERVICE special authority. In this way, only specific users will have the ability to perform a trace command, which would grant them access to sensitive data. The user must be

authorized to the command and have either *SERVICE special authority, or be authorized to the Service Trace function of the operating system through iSeries Navigator's Application Administration support. The Change Function Usage Information (QSYSCHFUI) API, with the function ID of QIBM_SERVICE_TRACE, can also be used to change the list of users that are allowed to perform trace operations.

The commands to which access can be granted in this way include:

Table 60.

STRCMNTRC	Start Communications Trace
ENDCMNTRC	End Communications Trace
PRTCMNTRC	Print Communications Trace
DLTCMNTRC	Delete Communications Trace
CHKCMNTRC	Check Communications Trace
TRCCNN	Trace Connection (see"Granting Access to Traces")
TRCINT	Trace Internal
STRTRC	Start Job Trace
ENDTRC	End Job Trace
PRTTRC	Print Job Trace
DLTTRC	Delete Job Trace

Granting Access to Traces: Trace commands, such as TRCCNN (Trace Connection) are powerful commands that should not be granted to all users who need access to other service and debug tools. Following the steps below will let you limit who can access these trace commands without having *SERVICE authority:

1. In iSeries Navigator, open Users and Groups.
2. Select All Users to view a list of user profiles.
3. Right-click the user profile to be altered.
4. Select Properties.
5. Click Capabilities.
6. Open the Applications tab.
7. Select Access for.
8. Select Host Applications.
9. Select Operating System.
10. Select Service.
11. Use the checkbox to grant or revoke access to trace command.

*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 CHGOBJAUT, CHGDLOAUD, and CHGAUD 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.

Note: Only a user with *ALLOBJ, *SECADM, and *AUDIT special authorities can give another user *AUDIT special authority.

*IOSYSCFG Special Authority

System configuration (*IOSYSCFG) special authority gives the user the ability to change how the system is configured. For example, adding or removing communications configuration information, working with TCP/IP servers, and configuring the internet connection server (ICS). Most commands for configuring communications require *IOSYSCFG special authority. Appendix D shows what special authorities are required for specific commands.

Note: You need *ALLOBJ to be able to change data using service functions.

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, you should control the following situations for user profiles and programs:

- Whether user profiles with special authorities can be used to submit jobs
- Whether programs created by these users can run using the authority of the program owner.

Programs adopt the *ALLOBJ special authority of the owner if:

- The programs are created by users who have *ALLOBJ special authority
- The user specifies USRPRF(*OWNER) parameter on the command that creates the program.

How LAN Server Uses Special Authorities

The LAN Server licensed program uses the special authorities in a user's profile to determine what operator capabilities the user should have in a LAN server environment. Following are the operator capabilities the system gives to LAN server users:

***ALLOBJ**
: System administrator

***IOSYSCFG**
: Server resource operator privilege

***JOBCTL**
: Communication device operator privilege

***SECADM**
: Accounts operator privilege

***SPLCTL**
: Print operator privilege

- *SAVSYS special authority applies when you save information using the /QFPNWSSTG directory. *SAVSYS special authority does apply when saving objects using the /QLANSrv directory, you must have the necessary permission (authority) to the object or LAN administrator authority.

- *ALLOBJ special authority gives enough authority to save /QLANSrv objects and their authority information if both of the following are true:
  - You are a defined user in the LAN domain.
  - The domain controller is a File Server I/O Processor on the local iSeries system.

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 iSeries, 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 2 on page 81.

Table 61. 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 iSeries 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 iSeries 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 iSeries applications.

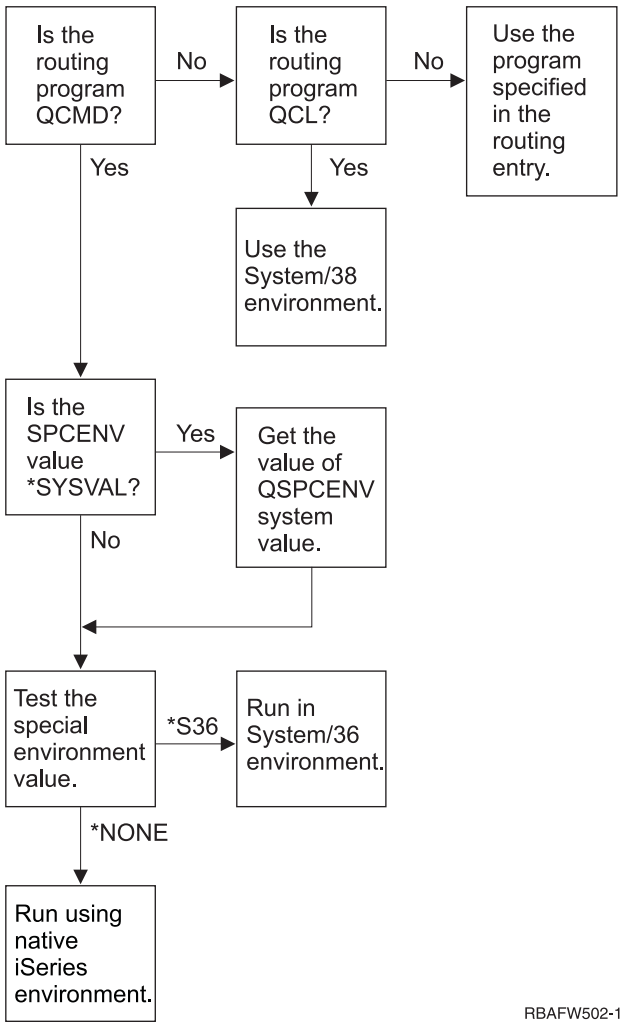


Figure 2. Description of Special Environment

Description of Special Environment

Special environment determines the environment the user operates in after signing on. The user can operate in the iSeries, 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. The following description explains Figure 2.

The system determines if the routing program is QCMD. If it is not, then the system checks to see if the routing program is QCL. If the routing program is QCL, then the system will use the System/38 special environment. If the routing program is not QCL, then the system uses the program specified in the routing entry.

If the routing program is QCMD, then the system determines if the SPCENV system value is set. If it is set then the system retrieves the value for QSPCENV system value and the system tests the special environment value. If SPCENV system value is not set, then the system tests the special environment value.

If the special environment value is set to *S36, the system runs the System/36 special environment. If the special environment value is set to *NONE, then the system runs the native iSeries 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 3 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 3. Sign-On Information Display

Table 62. 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 1 on page 68 shows an example of the password expiration message.

Recommendations: Use the user profile password interval to require profiles with *SERVICE, *SAVSYS, or *ALLOBJ special authorities to change passwords more frequently than other users.

Table 63. 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.

Table 64. 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 user profiles. 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.

Table 65. 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.

The maximum storage value is independently applied to each independent auxiliary storage pool (ASP) on the system. Therefore, specifying a value of 5000 means that the user profile can use the following:

- 5000 KB of auxiliary storage in the system ASP and basic user ASPs.
- 5000 KB of auxiliary storage in independent ASP 00033 (if it exists).
- 5000 KB of auxiliary storage in independent ASP 00034 (if it exists).

This provides a total of 15,000 KB of auxiliary storage from the whole system.

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.

Table 66. 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 kilobyte 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.

Table 67. 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 and *EXECUTE 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* book for more information about job descriptions and their uses.

Table 68. 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.

Table 69. 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, *OBJOPR, *READ, *ADD, *UPD, and *DLT authority to the group profile.

Notes: 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 135.

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. You may specify up to 15 additional groups for the user in the *Supplemental group profile* (SUPGRPPRF) parameter.

When a group profile is specified in a user profile, the user is automatically granted *OBJMGT, *OBJOPR, *READ, *ADD, *UPD, and *DLT authorities to the group profile, if the group profile is not already one of the user's group profiles. These authorities are necessary for system functions and should not be removed.

If a profile specified in the GRPPRF parameter is not already a group profile, the system sets information in the profile marking it as a group profile. The system also generates a gid for the group profile, if it does not already have one.

See "Planning Group Profiles" on page 229 for more information about using group profiles.

Table 70. 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 use the *owner* parameter in the user profile to specify who owns any new objects created by the user. Objects can be owned either by the user or by the user's first group (the value of the GRPPRF parameter). You can specify the *OWNER* field only if you have specified the *Group profile* field.

Table 71. 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. **Notes:** 1. If you give ownership to the group, all members of the group can change, replace, and delete the object. 2. The *GRPPRF parameter is ignored for all file systems except QSYS.LIB. In cases where the parameter is ignored, the user retains ownership of the object.

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. Group authority applies to the profile specified in the GRPPRF parameter. It does not apply to supplemental group profiles specified in the SUPGRPPRF parameter.

Table 72. Possible Values for GRPAUT:

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

See "Defining How Information Can Be Accessed" on page 120 for a complete explanation of the authorities that can be granted.

Group Authority Type

Add User prompt:
Not shown

CL parameter:
GRPAUTTYP

Length:
10

When a user creates a new object, the *Group authority type* parameter in the user's profile determines what type of authority the user's group receives to the new object. The GRPAUTTYP parameter works with the OWNER, GRPPRF, and GRPAUT parameters to determine the group's authority to a new object.

Table 73. Possible Values for GRPAUTTYP: [1]

***PRIVATE**	The authority defined in the GRPAUT parameter is assigned to the group profile as a private authority.
***PGP**	The group profile defined in the GRPPRF parameter is the primary group for the newly created object. The primary group authority for the object is the authority specified in the GRPAUT parameter.

[1] Private authority and primary group authority provide the same access to the object, but they may have different performance characteristics. "Primary Group for an Object" on page 130 explains how primary group authority works.

Recommendations: Specifying *PGP is a method for beginning to use primary group authority. Consider using GRPAUTTYP(*PGP) for users who frequently create new objects.

Supplemental Groups

Add User prompt:
Not shown

CL parameter:
SUPGRPPRF

Length:
150

Authority:
To specify supplemental groups when creating or changing a user profile, you must have *OBJMGT, *OBJOPR, *READ, *ADD, *UPD, and *DLT authority to each group profile.

Note: *OBJMGT authority cannot come from adopted authority. For more information, see "Objects That Adopt the Owner's Authority" on page 135.

You may specify the names of up to 15 profiles from which this user is to receive authority. The user becomes a member of each supplemental group profile. The user cannot have supplemental group profiles if the GRPPRF parameter is *NONE.

When supplemental group profiles are specified in a user profile, the user is automatically granted *OBJMGT, *OBJOPR, *READ, *ADD, *UPD, and *DLT authorities to each group profile, if the group profile is not already one of the user's group profiles. These authorities are necessary for system functions and should not be removed. If a profile specified in the SUPGRPPRF parameter is not already a group profile, the system sets information in the profile marking it as a group profile. The system also generates a gid for the group profile, if it does not already have one.

See "Planning Group Profiles" on page 229 for more information about using group profiles.

Table 74. Possible Values for SUPGRPPRF

***NONE**	No supplemental groups are used with this user profile.
group- profile- name	Specify up to 15 names of group profiles to be used with this user profile. These profiles, in addition to the profile specified in the GRPPRF parameter, are used to give the user access to objects.

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* book for more information about job accounting.

Table 75. 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.

Table 76. 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. *EXECUTE for the message queue library.

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.

Table 77. 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.

Table 78. 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

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.

Table 79. 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

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.

Table 80. 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 *Printer Device Programming* book.

Table 81. 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 *EXECUTE 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 *Printer Device Programming* book.

Table 82. 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.

Table 83. 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

*EXECUTE 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 185 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.

Table 84. 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 Attention Program (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.

Table 85. 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 *EXECUTE 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.

Table 86. 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.

Table 87. 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.

Table 88. 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 or Region Identifier

Add User prompt:
Not shown

CL parameter:
CNTRYID

Length:
10

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

Table 89. Possible Values for CNTRYID:

***SYSVAL**	The system value QCNTRYID is used to determine the country or region identifier.
country or region identifier	Specify the country or region 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.

Table 90. 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.

Character Identifier Control

Add User prompt:
Not shown

CL parameter:
CHRIDCTL

Length:
10

The *CHRIDCTL* attribute controls the type of coded character set conversion that occurs for display files, printer files and panel groups. The character identifier control information from the user profile is used only if the *CHRIDCTL special value is specified on the CHRID command parameter on the create, change, or override commands for display files, printer files, and panel groups.

Table 91. Possible Values for CHRIDCTL:

***SYSVAL**	The system value QCHRIDCTL is used to determine the character identifier control.
***DEVD**	The CHRID of the device is used to represent the CCSID of the data. No conversions occur, since the CCSID of the data is always the same as the CHRID of the device.
***JOBCCSID**	Character conversion occurs when a difference exists between the device CHRID, job CCSID, or data CCSID values. On input, character data is converted from the device CHRID to the job CCSID when it isnecessary. On output, character data is converted from the job CCSID to the device CHRID when it is necessary. On output, character data is converted from the file or panel group CCSID to the device CHRID when it is necessary.

Job Attributes

Add User prompt:
Not shown

CL parameter:
SETJOBATR

Length:
160

The *SETJOBATR* field specifies which job attributes are to be taken at job initiation from the locale specified in the LOCALE parameter.

Table 92. Possible Values for SETJOBATR:

***SYSVAL**	The system value QSETJOBATR is used to determine which job attributes are to be taken from the locale.
***NONE**	No job attributes are to be taken from the locale.
	Any combination of the following values may be specified:
***CCSID**	The coded character set identifier from the locale is used. The CCSID value from the locale will override the user profile CCSID.
***DATFMT**	The date format from the locale is used.
***DATSEP**	The date separator from the locale is used.
***DECFMT**	The decimal format from the locale is used.
***SRTSEQ**	The sort sequence from the locale is used. The sort sequence from the locale will override the user profile sort sequence.
***TIMSEP**	The time separator from the locale is used.

Locale

Add User prompt:
Not shown

CL parameter:
LOCALE

Length:
2048

The *LOCALE* field specifies the path name of the locale that is assigned to the LANG environment variable for this user.

Table 93. Possible Values for LOCALE:

***SYSVAL**	The system value QLOCALE is used to determine the locale path name to be assigned for this user.
***NONE**	No locale is assigned for this user.
***C**	The C locale is assigned for this user.
***POSIX**	The POSIX locale is assigned for this user.
locale path name	The path name of the locale to be assigned to 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.

Table 94. 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). "Authority Displays" on page 141 shows an example of the expert version of the display.
***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.

User Identification Number

Add User prompt:
Not shown

CL parameter:
UID

Length:
10,0

The integrated file system uses the user identification number (uid) to identify a user and verify the user's authority. Every user on the system must have a unique uid.

Table 95. Possible Values for UID:

***GEN**	The system generates a unique uid for this user. The generated uid will be greater than 100.
uid	A value from 1 to 4294967294 to be assigned as the uid for this user. The uid must not be already assigned to another user.

Recommendations: For most installations, let the system generate a uid for new users by specifying UID(*GEN). However, if your system is part of a network, you may need to assign uids to match those assigned on other systems in the network. Consult your network administrator.

Group Identification Number

Add User prompt:
Not shown

CL parameter:
GID

Length:
10,0

The integrated file system uses the group identification number (gid) to identify this profile as a group profile. A profile that is used as a group profile by the integrated file system must have a gid.

Table 96. Possible Values for GID:

***NONE**	This profile does not have a gid.
***GEN**	The system generates a unique gid for this profile. The generated gid will be greater than 100.
gid	A value from 1 to 4294967294 to be assigned as the gid for this profile. The gid must not be already assigned to another profile.

Recommendations: For most installations, let the system generate a gid for new group profiles by specifying GID(*GEN). However, if your system is part of a network, you may need to assign gids to match those assigned on other systems in the network. Consult your network administrator.

Do not assign a gid to a user profile that you do not plan to use as a group profile. In some environments, a user who is signed on and has a gid is restricted from performing certain functions.

Home Directory

Add User prompt:
Not shown

CL parameter:
HOMEDIR

Length:
2048

The home directory is the user's initial working directory for the integrated file system. The home directory is the user's current directory if a different current directory has not been specified. If the home directory specified in the profile does not exist when the user signs on, the user's home directory is the root (/) directory.

Table 97. Possible Values for HOMEDIR:

***USRPRF**	The home directory assigned to the user is `/home/xxxxx`, where `xxxxx` is the user's profile name.
home-directory	The name of the home directory to assign to this 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

Table 98. 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 120 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.

Table 99. 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 100 on page 102 shows how the OBJAUD values for the user and the object work together:

Table 100. 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 263 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.

Table 101. 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**	Save and 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 253 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
- Primary group 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 236 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 3: How User Authority to an Object Is Checked" on page 161 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 153 for more information.

Primary Group Authorities

The names of all the objects for which the profile is the primary group are stored with the group profile. You can display the objects for which the profile is the primary group using the DSPUSRPRF command: `DSPUSRPRF` *`group-profile-name`* `TYPE(*OBJPGP)`. You can also use the Work with Objects by Primary Group (WRKOBJPGP) command.

Owned Object Information

Private authority information for an object is 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. The size of an owner profile affects performance when displaying and working with the authority to owned objects, and when saving or restoring profiles. System operations can also be impacted. To prevent impacts to either performance or system operations, distribute ownership of objects to multiple profiles. Because the size of a user profile can impact your performance, it is suggested that you do not assign all (or nearly all) objects to only one owning profile.

Digital ID Authentication

The iSeries security infrastructure allows x.509 digital certificates to be used for identification. The digital certificates allow users to secure communications and ensure message integrity.

The digital ID APIs create, distribute, and manage digital certificates associated with user profiles. See the API topic in the Information Center (see "Prerequisite and related information" on page xvi) for details about the following APIs:

- Add User Certificate (QSYADDUC)
- Remove User Certificate (QSYRMVUC)
- List User Certificate (QSYLSTUC)
- Find Certificate User (QSYFNDUC)
- Add Validation List Certificate (QSYADDVC)
- Remove Validation List Certificate (QSYRMVVC)
- List Validation List Certificate (QSYLSTVC)
- Check Validation List Certificate (QSYCHKVC)
- Parse Certificate (QSYPARSC)

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 create, change, or delete 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.
- Using the iSeries Navigator display from the iSeries Access folder.

The user who creates the user profile owns it and is given *ALL authority to it. The user profile is given *OBJMGT and *CHANGE authority to itself. These authorities are necessary for normal operations and should not be removed.

A user profile cannot be created with more authorities or capabilities than those of the user who creates the profile.

Note: When you perform a CRTUSRPRF, you can not create a user profile (*USRPRF) into an independent disk pool. However, when a user is privately authorized to an object in the independent disk pool, is the owner of an object on an independent disk pool, or is the primary group of an object on an independent disk pool, the name of the profile is stored on the independent disk pool. If the independent disk pool is moved to another system, the private authority, object ownership, and primary group entries will be attached to the profile with the same name on the target system. If a profile does not exist on the target system, a profile will be created. The user will not have any special authorities and the password will be set to *NONE.

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. When you use the WRKUSRPRF command with *BASIC assistance level, you

will access the Work with User Enrollment display. If *INTERMED assistance level is specified, you will access the Work with User Profiles display.

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  . . . . . .   NEWUSER1
Set password to expired  . . . .   *YES
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 add the user to the system directory.

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. 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.

The main menu also includes user Options 51—59. These additional options (Options 51--59) are processed similar to Option 50, except the default values for the following fields are blank:

- Text for menu options
- User program
- Library

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:

Home directory
*USRPRF

Locale job attributes
Locale job attributes

Locale Locale

User profile
Blank. Must be filled in.

Password
*USRPRF

Message queue
*USRPRF

Document password
*NONE

User Identification Number
*GEN

Group Identification Number
*NONE

Authority
*EXCLUDE

You can change any fields on the Create User Profile display. Private authorities of the copy-from profile are not copied. In addition, internal objects containing user preferences and other information about the user will not be 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. Limited to 8 characters.

Password
Blank. If you do not enter a value, the profile is created with the password equal to the default value specified for the PASSWORD parameter of the CRTUSRPRF command.

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

User Identification Number
*GEN

Group Identification Number
*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 153 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.

A user cannot change a user profile to have more special authorities or capabilities than the user who changes the profile.

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.

You cannot delete a user profile if it is the primary group for any objects. When you use the intermediate assistance level to delete a user profile, you can change or remove the primary group for objects. You can use the DSPUSRPRF command with the *OBJPGP (object primary group) option to list any objects for which a profile is the primary group.

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
- All objects for which the profile is the primary group.

```
                      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
Primary group option:
  Primary group value  . . . . .   *NOCHG        *NOCHG, *PGP
  New primary group  . . . . . .
  New primary group authority  .
```

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. You can change the primary group for all objects for which the group profile is the primary group. If you want to handle objects individually, you can use the Work with Objects by Primary Group (WRKOBJPGP) command. The displays for both commands are similar:

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

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.
3. Any objects for which the deleted user profile was the primary group will have a primary group of *NONE.

Working with Objects by Primary Group

You can use the Work with Objects by Primary Group (WRKOBJPGP) command to display and work with objects for which a profile is the primary group. You can use this display to change an object's primary group to another profile or to set it's primary group to *NONE.

```
                     Work with Objects by Primary Group

Primary group  . . . . . :   DPTAR

Type options, press Enter.
  2=Edit authority        4=Delete   5=Display authority
  8=Display description   9=Change primary group
                                                    ASP
Opt    Object     Library    Type   Attribute     Device
       CUSTMAST   CUSTLIB    *FILE               *SYSBAS
       CUSTWRK    CUSTLIB    *FILE               *SYSBAS
       CUSTLIB    QSYS       *LIB                *SYSBAS
```

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     08/04/0x            Anders, Roger
            VINCENT     09/15/0x            Vincent, Mark
DPTWH
            ANDERSR     08/04/0x            Anders, Roger
            HOGANR      09/06/0x            Hogan, Richard
            QUINN       09/06/0x            Quinn, Rose
QSECOFR
            JONESS      09/20/0x            Jones, Sharon
            HARRISON    08/29/0x            Harrison, Ken
*NO GROUP
            DPTSM       09/05/0x     X      Sales and Marketing
            DPTWH       09/18/0x     X      Warehouse
```

By pressing F11, you are able to see which user profiles have passwords defined for use at the various password levels.

```
                        Display Authorized Users

                    Password  Password   Password   Password
User      Group     Last      for level  for level     for
Profile   Profile   Changed    0 or 1     2 or 3    NetServer
ANGELA              04/21/0x    *YES       *NO        *YES
ARTHUR              07/07/0x    *YES       *YES       *YES
CAROL1              05/15/0x    *YES       *YES       *YES
CAROL2              05/15/0x    *NO        *NO        *NO
CHUCKE              05/18/0x    *YES       *NO        *YES
DENNISS             04/20/0x    *YES       *NO        *YES
DPORTER             03/30/0x    *YES       *NO        *YES
GARRY               08/04/0x    *YES       *YES       *YES
JANNY               03/16/0x    *YES       *NO        *YES
```

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 277 gives suggestions for reports.

Types of User Profile Reports

The following commands provide user profile reports.

- Print User Profile (PRTUSRPRF)

  This command allows you to print a report containing information for the user profiles on the system. Four different reports can be printed. One contains authority type information, one contains environment type information, one contains password type information, and one contains password level type information.
- Analyze Default Password (ANZDFTPWD)

  This command allows you to print a report of all the user profiles on the system that have a default password and to take an action against the profiles. A profile has a default password when the user profile name matches the profile's password.

  User profiles on the system that have a default password can be disabled and their passwords can be set to expired.

Renaming a User Profile

The system does not provide a direct method for renaming a user profile.

A new profile can be created with the same authorities for a user with a new name. Some information, however, cannot be transferred to the new profile. The following are examples of information that cannot be transferred:

- Spool files.
- Internal objects containing user preferences and other information about the user will be lost.
- Digital certificates that contain the user name will be invalidated.
- The uid and gid information retained by the IFS cannot be changed.
- You may not be able to change the information that is stored by applications that contain the user name.

Applications that are run by the user can have ″application profiles″. Creating a new iSeries user profile to rename a user does not rename any application profiles the user may have. A Lotus Notes profile is one example of an application profile.

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.
2. Give JONESM all the private authorities of SMITHM using the Grant User Authority (GRTUSRAUT) command:

```
GRTUSRAUT JONESM REFUSER(SMITHM)
```

3. Change the primary group of all objects that SMITHM is the primary group of using the Work with Objects by Primary Group (WRKOBJPGP) command:

   ```
   WRKOBJPGP PGP(SMITHM)
   ```

   Enter option 9 on all objects that need their primary group changed and enter NEWPGP (JONESM) on the command line.

   Note: JONESM must have a gid assigned using the GID parameter on the Create or Change User Profile (CRTUSRPRF or CHGUSRPRF) command.

4. Display the SMITHM user profile using the Display User Profile (DSPUSRPRF) command:

   ```
   DSPUSRPRF USRPRF(SMITHM)
   ```

   Write down the uid and gid for SMITHM.

5. 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.

6. Change the uid and the gid of JONESM to the uid and gid that belonged to SMITHM by using the Change User Profile (CHGUSRPRF) command:

   ```
   CHGUSRPRF USRPRF(JONESM) UID(uid from SMITHM)
             GID(gid from SMITHM)
   ```

   If JONESM owns objects in a directory, the CHGUSRPRF command cannot be used to change the uid and gid. Use the QSYCHGID API to change the uid and gid of user profile JONESM.

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 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 53 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.

User Profile Exit Points

Exit points are provided to create, change, delete, or restore user profiles. You can write your own exit programs to perform specific user profile functions. When you register your exit programs with any of the user profile exit points, you are notified when a user profile is created, changed, deleted, or restored. At the time of notification, your exit program can perform any of the following:

- Retrieve information about the user profile
- Enroll the user profile that was just created in the system directory.
- Create necessary objects for the user profile.

For more information about the Security exit programs, see the API topic in the Information Center (see "Prerequisite and related information" on page xvi for details).

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.

IBM-supplied user profiles, except QSECOFR, are shipped with a password of *NONE and are not intended for sign-on. To allow you to install your system the first time, the password for the security officer (QSECOFR) profile is the same for every system that is shipped. However, the password for QSECOFR is shipped as expired. For new systems, you are required to change the password the first time you sign-on as QSECOFR.

When you install a new release of the operating system, passwords for IBM-supplied profiles are not changed. If profiles such as QPGMR and QSYSOPR have passwords, those passwords are not set to *NONE automatically.

Appendix B, "IBM-Supplied User Profiles" on page 291 contains a complete list of all the IBM-supplied user profiles and the field values for each profile.

Note: IBM-supplied profiles are provided, but they are used by the Operating System/400. Therefore, signing on with these profiles or using the profiles to own user (non-IBM supplied) objects is **not** recommended.

Changing Passwords for IBM-Supplied User Profiles

If you need to sign on with one of the IBM-supplied profiles, you can change the password using the CHGUSRPRF command. You can also change these passwords using an option from the SETUP menu. To protect your system, you should leave the password set to *NONE for all IBM-supplied profiles except QSECOFR. Do not allow trivial passwords for the QSECOFR profile.

```
                  Change Passwords for IBM-Supplied

Type new password below for IBM-supplied user,
type password again to verify  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) . . . . . .
```

Working with service tools user IDs

There are several enhancements and additions to service tools for this release that make them easier to use and understand. System service tools (SST) You can now manage and create service tools user IDs from system service tools (SST) by selecting option 8 (Work with service tools user IDs) from the main SST display. You no longer need to go into dedicated service tools (DST) to reset passwords,

grant or revoke privileges, or create service tools user IDs. **Note**: Information regarding Service tools has been moved to the Information Center.

- **Password management enhancements**

  The server is shipped with limited ability to change default and expired passwords. This means that you cannot change service tools user IDs that have default and expired passwords through the Change Service Tools User ID (QSYCHGDS) API, nor can you change their passwords through SST. You can only change a service tools user ID with a default and expired password through DST. And, you can change the setting to allow default and expired passwords to be changed. Also, you can use the new Start service tools (STRSST) privilege to create a service tools user ID that can access DST, but can be restricted from accessing SST.

- **Terminology changes**

  The textual data and other documentation have been changed to reflect the new service tools terminology. Specifically, the term service tools user IDs replaces previous terms, such as DST user profiles, DST user IDs, service tools user profiles, or variations of these names.

For information on how to work with Service tools, see the Information Center topic, Service tools (**Security—>Service tools**). See "Prerequisite and related information" on page xvi for more information on accessing the Information Center.

System Password

The system password is used to authorize system model changes, certain service conditions, and ownership changes. If these changes have occurred on your system, you may be prompted for the system password when you perform an IPL.

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 156 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.

Defining Who Can Access Information

You can give authority to individual users, groups of users, and the public.

Note: In some environments, a user's authority is referred to as a **privilege**.

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, object ownership, or primary group authority 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.

Primary Group Authority:

You can specify a primary group for an object and the authority the primary group has to the object. Primary group authority is stored with the object and may provide better performance than private authority granted to a group profile. Only a user profile with a group identification number (gid) may be the primary group for an object. Primary group authority 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.

Note: In some environments, the authority associated with an object is called the object's **mode of access**.

Authority to an object is divided into three 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.**Field Authority** defines what operations can be performed on the data fields.

Table 102 describes the types of authority available and lists some examples of how the authorities are used. In most cases, accessing an object requires a combination of object, data, field authorities. Appendix D provides information about the authority that is required to perform a specific function.

Table 102. Description of Authority Types

Authority	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. All functions defined for *OBJALTER and *OBJREF.
*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.
*OBJALTER	Object Alter	Add, clear, initialize and reorganize members of the database files. Alter and add attributes of database files: add and remove triggers. Change the attributes of SQL packages.
*OBJREF	Object Reference	Specify a database file as the parent in a referential constraint. For example, you want to define a rule that a customer record must exist in the CUSMAS file before an order for the customer can be added to the CUSORD file. You need *OBJREF authority to the CUSMAS file to define this rule.
*AUTLMGT	Authorization List Management	Add and remove users and their authorities from the authorization list [2].

Table 102. Description of Authority Types (continued)

Authority	Name	Functions Allowed
Data Authorities:		
*READ	Read	Display the contents of the object, such as viewing records in a file.
*ADD	Add	Add entries to an object, such as adding messages to a message 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.
*EXECUTE	Execute	Run a program, service program, or SQL package. Locate an object in a library or a directory.
Field Authorities:		
*Mgt	Management	Specify the security for the field.
*Alter	Alter	Change the attributes of the field.
*Ref	Reference	Specify the field as part of the parent key in a referential constraint.
*Read	Read	Access the contents of the field. For example, display the contents of the field.
*Add	Add	Add entries to data, such as adding information to a specific field.
*Update	Update	Change the content of existing entries in the field.

[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 127 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 103 shows the system-defined authorities available using the object authority commands and displays.

Table 103. System-Defined Authority

Authority	*ALL	*CHANGE	*USE	*EXCLUDE
Object Authorities				
*OBJOPR	X	X	X	
*OBJMGT	X			
*OBJEXIST	X			
*OBJALTER	X			
*OBJREF	X			
Data Authorities				
*READ	X	X	X	
*ADD	X	X		

Table 103. System-Defined Authority (continued)

Authority	*ALL	*CHANGE	*USE	*EXCLUDE
*UPD	X	X		
*DLT	X	X		
*EXECUTE	X	X	X	

Table 104 shows additional system-defined authorities that are available using the WRKAUT and CHGAUT commands:

Table 104. System-Defined Authority

Authority	*RWX	*RW	*RX	*R	*WX	*W	*X
Object Authorities							
*OBJOPR	X	X	X	X	X	X	X
*OBJMGT							
*OBJEXIST							
*OBJALTER							
*OBJREF							
Data Authorities							
*READ	X	X	X	X			
*ADD	X	X			X	X	
*UPD	X	X			X	X	
*DLT	X	X			X	X	
*EXECUTE	X		X		X		X

The LAN Server licensed program uses access control lists to manage authority. A user's authorities are called **permissions**. Table 105 shows how the LAN Server permissions map to object and data authorities:

Table 105. LAN Server Permissions

Authority	LAN Server Permissions
*EXCLUDE	None
Object Authorities	
*OBJOPR	See note 1
*OBJMGT	Permission
*OBJEXIST	Create, Delete
*OBJALTER	Attribute
*OBJREF	No equivalent
Data Authorities	
*READ	Read
*ADD	Create
*UPD	Write
*DLT	Delete
*EXECUTE	Execute

1 Unless NONE is specified for a user in the access control list, the user is implicitly given *OBJOPR.

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.
- Ensure that public authority is sufficient for all objects (in the library) that are used by applications (*USE or *CHANGE).
- 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. 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.

Attention: If a user is running under adopted authority when a library is added to the library list, the user remains authorized to the library 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.

In addition, applications that use library lists rather than qualified library names have a potential security exposure. A user who is authorized to the commands to work with library lists could potentially run a different version of a program. See "Library Lists" on page 193 for more information.

Field Authorities

Field authorities are now supported for database files. Authorities supported are Reference and Update. You can only administer these authorities through the SQL

statements, GRANT and REVOKE. You can display these authorities through the Display Object Authority (DSPOBJAUT) and the Edit Object Authority (EDTOBJAUT) commands. You can only display the field authorities with the EDTOBJAUT command; you cannot edit them.

```
                          Display Object Authority

Object . . . . . :   PLMITXT     Owner  . . . . . . . . :   PGMR1
  Library. . . . :   RLN         Primary group  . . . :   DPTAR
Object type. . . :   *FILE       ASP Device   . . . . :   *SYSBAS

Object secured by authorization list  . . . . . . . . . :   *NONE
                        Object    ---------------Data---------------
User        Group       Authority  Read  Add  Update  Delete  Execute
PGMR1                   *ALL        X     X     X       X       X
USER1                   *USE        X                           X
USER2                   USER DEF    X           X               X
USER3                   USER DEF    X                           X
*PUBLIC                 *CHANGE     X     X     X       X       X




Press Enter to continue


F3=Exit  F11=Nondisplay detail F12=Cancel F16=Display field authorities
```

Figure 4. Display Object Authority display showing F16=Display field authorities. This function key will be displayed when a database file has field authorities.

```
                          Display Field Authority

Object . . . . . :   PLMITXT         Owner  . . . . . :   PGMR1
  Library  . . . . . :     RLN           Primary group  . . . :   *NONE
Object type  . . . . :   *FILE

                           Object    -----Field Authorities-----------
Field        User       Authority  Mgt  Alter Ref  Read  Add  Update
Field3       PGMR1      *ALL        X      X   X    X     X      X
             USER1      *Use                        X
             USER2      USER DEF                    X            X
             USER3      USER DEF                X   X
             *PUBLIC    *CHANGE                     X     X      X
Field4       PGMR1      *ALL        X      X        X     X      X
             USER1      *Use                        X
             USER2      USER DEF                    X
             USER3      USER DEF                    X
             *PUBLIC    *CHANGE                     X     X      X
                                                                More
Press Enter to continue.

F3=Exit F5=Refresh F12=Cancel F16=Repeat position to F17=Position to
```

Figure 5. Display Field Authority display. When F17=Position to, is pressed the Position the List prompt will be displayed. If F16 is pressed, the previous position to operation will be repeated

Changes for field authorities include the following:

- The Print Private Authority (PRTPVTAUT) command has a new field that indicates when a file has field authorities.
- The Display Object Authority (DSPOBJAUT) command now has a new Authority Type parameter to allow display of object authorities, field authorities, or all authorities. If the object type is not *FILE, you can display only object authorities.
- Information provided by List Users Authorized to Object (QSYLUSRA) API now indicates if a file has field authorities.
- The Grant User Authority (GRTUSRAUT) command will not grant a user's field authorities.
- When a grant with reference object is performed using the GRTOBJAUT command and both objects (the one being granted to and the referenced one) are database files, all field authorities will be granted where the field names match.
- If a user's authority to a database file is removed, any field authorities for the user are also removed.

Security and the System/38 Environment

The System/38 Environment and CL programs of type CLP38 represent a potential security exposure. When a non-library qualified command is entered from the System/38 Command Entry screen, or invoked by any CLP38 CL program, library QUSER38 (if it exists) is the first library searched for that command. Library QSYS38 is the second library searched. A programmer or other knowledgeable user could place another CL command in either of these libraries and cause that command to be used instead of one from a library in the library list.

Library QUSER38 is not shipped with the operating system. However, it can be created by anyone with enough authority to create a library.

See the *System/38 Environment Programming* manual for more information about the System/38 Environment.

Recommendation for System/38 Environment

Use these measures to protect your system for the System/38 Environment and CL programs of type CLP38:

- Check the public authority of the QSYS38 library and if it is *ALL or *CHANGE then change it to *USE.
- Check the public authority of the QUSER38 library and if it is *ALL or *CHANGE then change it to *USE.
- If the QUSER38 and QSYS38 do not exist then create them and set them to public *USE authority. This will prevent anyone else from creating it at a later time and giving themselves or the public too much authority to it.

Directory Security

When accessing an object in a directory, you must have authority to all the directories in the path containing the object. You must also have the necessary authority to the object to perform the operation you requested.

You may want to use directory security in the same way that you use library security. Limit access to directories and use public authority to the objects within the directory. Limiting the number of private authorities defined for objects improves the performance of the authority checking process.

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. When you give a user authority to the authorization list, the operating system actually grants a **private authority for that user** to the authorization list.

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 227 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 the users' authority to the authorization list and change the authorities for those users. *AUTLMGT authority, by itself, does not 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 the authority for a user 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. Also, if an authorization list secures an object in QSYS and a complete system restore is required, the link between the objects in QSYS and the authorization list is lost. After you install a new release or restore your system, use the EDTOBJAUT or GRTOBJAUT command to re-establish the link between the IBM-supplied object and the authorization list.

The *Implementation Guide for AS/400 Security and Auditing* redbook contains sample programs, such as ALLAUTL and FIXAUTL, that can be used to attach authorization lists to the objects after the authorization lists are restored.

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)
```

"Assigning Authority and Ownership to New Objects" on page 131 shows more examples of how the system assigns ownership and authority to new objects.

Attention: 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.

Authority for New Objects in a Directory

When you create a new object in a directory, you specify the data authority and object authority that the public receives for the object. You use the *INDIR option to have the public authority set the same as that of the last directory in the path name.

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.

"Assigning Authority and Ownership to New Objects" on page 131 shows examples of how the system assigns ownership to new objects.

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. Ownership can be transferred to an individual user profile or a group profile. A group profile can own objects whether or not the group has members.

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 a profile that owns many objects affects performance when displaying and working with the authority to objects it owns, and when saving or restoring profiles. System operations can also be impacted. To prevent impacts to either performance or system operations, do not assign objects to only one owner profile for your entire iSeries system. Each application and the application objects should be owned by a separate profile. Also, IBM-supplied user profiles should not own user data or objects.

The owner of an object also needs sufficient storage for the object. See "Maximum Storage" on page 84 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 type* (GRPAUTTYP) field in the user profile determines whether the group 1) becomes the primary group for the object or 2) is given private authority to the object. "Assigning Authority and Ownership to New Objects" on page 131 shows several examples.

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 and the QAUDCTL system value includes *AUDLVL.

Primary Group for an Object

You can specify a primary group for an object. The name of the primary group profile and the primary group's authority to the object are stored with the object. Using primary group authority may provide better performance than private group authority when checking authority to an object.

A profile must be a group profile (have a gid) to be assigned as the primary group for an object. The same profile cannot be the owner of the object and its primary group.

When a user creates a new object, parameters in the user profile control whether the user's group is given authority to the object and the type of authority given. The *Group authority type* (GRPAUTTYP) parameter in a user profile can be used to make the user's group the primary group for the object. "Assigning Authority and Ownership to New Objects" on page 131 shows examples of how authority is assigned when new objects are created.

Use the Change Object Primary Group (CHGOBJPGP) command or the Work with Objects by Primary Group (WRKOBJPGP) command to specify the primary group for an object. You can change the authority the primary group has using the Edit Object Authority display or the grant and revoke authority commands.

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 (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 17 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. QDFTOWN user profile is intended for system use only. You should not design your security such that QDFTOWN normally owns object.

Assigning Authority and Ownership to New Objects

The system uses several values to assign authority and ownership when a new object is created on the system:

Parameters on the CRTxxx command

The QCRTAUT system value

The CRTAUT value of the library

Values in the user profile of the creator

Figure 6 through Figure 9 show several examples of how these values are used:

QCRTAUT system value:
*CHANGE

CRTAUT library parameter:
*USE

Values in USERA (Creator) Profile:

GRPPRF:
DPT806

OWNER:
*USRPRF

GRPAUT:
*CHANGE

GRPAUTTYP:
*PRIVATE

Command Used to Create Object:

```
CRTDTAARA DTAARA(CUSTLIB/DTA1)
        TYPE(*CHAR) AUT(*LIBCRTAUT)
```

or

```
CRTDTAARA DTAARA(CUSTLIB/DTA1)
        TYPE(*CHAR)
```

Values for New Object:

Public authority:
*USE

Owner authority:
USERA *ALL

Primary group authority:
None

Private authority:
DPT806 *CHANGE

Note: *LIBCRTAUT is the default value for the AUT parameter on most CRTxxx commands.

Figure 6. New Object Example: Public Authority from Library, Group Given Private Authority

QCRTAUT system value:
*CHANGE

CRTAUT library parameter:
*SYSVAL

Values in USERA (Creator) Profile:

GRPPRF:
DPT806

OWNER:
*USRPRF

GRPAUT:
*CHANGE

GRPAUTTYP:
*PRIVATE

Command Used to Create Object:

```
CRTDTAARA DTAARA(CUSTLIB/DTA1)
        TYPE(*CHAR) AUT(*LIBCRTAUT)
```

Values for New Object:

Public authority:
*CHANGE

Owner authority:
USERA *ALL

Primary group authority:
None

Private authority:
DPT806 *CHANGE

Figure 7. New Object Example: Public Authority from System Value, Group Given Private Authority

QCRTAUT system value:
*CHANGE

CRTAUT library parameter:
*USE

Values in USERA (Creator) Profile:

GRPPRF:
DPT806

OWNER:
*USRPRF

GRPAUT:
*CHANGE

GRPAUTTYP:
*PGP

Command Used to Create Object:

```
CRTDTAARA DTAARA(CUSTLIB/DTA1)
        TYPE(*CHAR) AUT(*LIBCRTAUT)
```

Values for New Object:

Public authority:
*USE

Owner authority:
USERA *ALL

Primary group authority:
DPT806 *CHANGE

Private authority:
None

Figure 8. New Object Example: Public Authority from Library, Group Given Primary Group Authority

QCRTAUT system value:
*CHANGE

CRTAUT library parameter:
*USE

Values in USERA (Creator) Profile:

GRPPRF:
DPT806

OWNER:
*GRPPRF

GRPAUT:

GRPAUTTYP:

Command Used to Create Object:

```
CRTDTAARA DTAARA(CUSTLIB/DTA1)
        TYPE(*CHAR) AUT(*CHANGE)
```

Values for New Object:

Public authority:
*CHANGE

Owner authority:
DPT806 *ALL

Primary group authority:
None

Private authority:
None

Figure 9. New Object Example: Public Authority Specified, Group Owns Object

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, *SQLPKG and Java programs 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.

Consult the Information Center concerning security considerations and adopted authority when using SQL packages (see "Prerequisite and related information" on page xvi for details).

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. Owner authority, primary group authority, and private authority are used for adopted authority.
  - Only the authority is adopted. No other user profile attributes are adopted. For example, the limited capabilities attributes are not adopted.
- 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 10. 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 139 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 11. 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.
- When a trigger program or exit point program is called, adopted authority from previous programs in the call stack will not be used as a source of authority for the trigger program or exit point program.
- The program adopt function is not used when you use the Change Job (CHGJOB) command to change the output queue for a job. The user profile making the change must have authority to the new output queue.
- 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 the owner of the program can specify REPLACE(*YES) on the CRTxxxPGM command when USRPRF(*OWNER) is specified on the original program.
- 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. The Print Adopting Objects (PRTADPOBJ) command provides a report with more information about objects that adopt authority. This command also provides an option to print a report for objects that changed since the last time the command was run.

"Flowchart 8: How Adopted Authority Is Checked" on page 169 provides more information about adopted authority. The topic "Using Adopted Authority in Menu Design" on page 218 shows an example of how to use adopted authority in an application.

Adopted Authority and Bound Programs:

An ILE* program (*PGM) is an object that contains one or more modules. It is created by an ILE* compiler. An ILE program can be bound to one or more service programs (*SRVPGM).

To activate an ILE program successfully, the user must have *EXECUTE authority to the ILE program and to all service programs to which it is bound. If an ILE program uses adopted authority from a program higher in the program call stack, that adopted authority **is** used to check authority to all service programs to which the ILE program is bound. If the ILE program adopts authority, the adopted authority will not be checked when the system checks the user's authority to the service programs at program activation time.

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 special 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.
- Programs that adopt authority and call other programs must perform a library qualified call. Do not use the library list (*LIBL) on the call.
- 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 from previous programs in the stack. If you do not want the program to use adopted authority, you can change the program with the Change Program (CHGPGM) command 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 220 shows an example of how to use this parameter in menu design. See "Use Adopted Authority (QUSEADPAUT)" on page 34 for information on the QUSEADPAUT system value.

Attention: In some situations, you can use the MODINVAU MI instruction to prevent passing adopted authority to called functions. The MODINVAU instruction can be used to prevent passing any adopted authority from C and C++ programs to called functions in another program or service program. This may be useful when you do not know the USEADPAUT setting of the function that is called.

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:

- Authority holders can only secure files in the system auxiliary storage pool (ASP) or a basic user ASP. They cannot secure files in an independent ASP.
- 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 and logical files created in the S/36 environment.
- 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 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, QTEMP, and QSPL0002 – QSPL0032.

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.

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, "Security Commands" on page 283 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.

Authority Displays

Four displays show object authorities:

Display Object Authority display

Edit Object Authority display

Display Authority display

Work with Authority display

This section describes some characteristics of these displays. Figure 12 shows the basic version of the Display Object Authority display:

```
                          Display Object Authority

Object . . . . . :   CUSTNO    Owner  . . . . . :   PGMR1
  Library. . . . . :   CUSTLIB   Primary group  . . . :   DPTAR
Object type  . . . :   *DTAARA   ASP device   . . . . :   *SYSBAS

Object secured by authorization list  . . . . . :   *NONE

                         Object
User        Group        Authority
PGMR1                    *ALL
DPTAR                    *CHANGE
DPTSM                    *USE
*PUBLIC                  *EXCLUDE
F3=Exit F11=Display detail object authorities F12=Cancel  F17=Top
```

Figure 12. Display Object Authority Display

The system-defined names of the authorities are shown on this display. F11 acts as a toggle between this and two other versions of the display. One shows detailed object authorities:

```
                          Display Object Authority

Object . . . . . :   CUSTNO          Owner . . . . . :   PGMR1
  Library. . . . . :   CUSTLIB       Primary group . . . :   DPTAR
Object type. . . . :   *DTAARA         ASP device  . . . :   *SYSBAS

Object secured by authorization list   . . . . . :   *NONE

                        Object    ----------Object-----------
User        Group       Authority  Opr  Mgt  Exist  Alter  Ref
PGMR1                   *ALL        X    X     X      X     X
DPTAR                   *CHANGE     X
DPTSM                   *USE        X
*PUBLIC                 *EXCLUDE    X
   :
F3=Exit F11=Display data authorities F12=Cancel F17=Top F18=Bottom
```

The other shows data authorities:

```
                          Display Object Authority

Object . . . . . :   CUSTNO      Owner  . . . . . :   PGMR1
  Library. . . . . :   CUSTLIB     Primary group  . . . :   DPTAR
Object type. . . . :   *DTAARA     ASP device   . . . . :   *SYSBAS

Object secured by authorization list. . . . . :   *NONE

                        Object     ---------------Data---------------
User        Group       Authority  Read  Add  Update  Delete  Execute
PGMR1                   *ALL        X     X     X       X       X
DPTAR                   *CHANGE     X     X     X       X       X
DPTSM                   *USE        X                           X
*PUBLIC                 *EXCLUDE
```

If you have *OBJMGT authority to an object, you see all private authorities for that object. If you do not have *OBJMGT authority, you see only your own sources of authority for the object.

For example, if USERA displays authority for the CUSTNO data area, only public authority is shown.

If USERB, who is a member of the DPTAR group profile, displays the authority for the CUSTNO data area, it looks like this:

```
                          Display Object Authority

Object . . . . . :   CUSTNO      Owner  . . . . . :   PGMR1
  Library. . . . . :   CUSTLIB     Primary group  . . . :   DPTAR
Object type. . . . :   *DTAARA     ASP device   . . . . :   *SYSBAS

Object secured by authorization list. . . . . :   *NONE

                       Object
User        Group      Authority
*GROUP      DPTAR      *CHANGE
```

If USERB runs a program that adopts the authority of PGMR1 and displays the authority for the CUSTNO data area, it looks like this:

```
                          Display Object Authority

Object .. . . . . :   CUSTNO      Owner  . . . . . :   PGMR1
  Library   . . . . :   CUSTLIB     Primary group  . . . :   DPTAR
Object type. . . . :   *DTAARA     ASP device   . . . . :   *SYSBAS

Object secured by authorization list . . . . . :   *NONE

                       Object
User        Group      Authority
PGMR1                  *ALL
*GROUP      DPTAR      *CHANGE
DPTSM                  *USE
*PUBLIC                *EXCLUDE
*ADOPTED               USER DEF
```

The *ADOPTED authority indicates only the additional authority received from the program owner. USERB receives from PGMR1 all the authorities that are not included in *CHANGE. The display shows all private authorities because USERB has adopted *OBJMGT. The detailed display looks like this:

```
                          Display Object Authority

Object . . . . . :   CUSTNO      Owner  . . . . . :   PGMR1
  Library. . . . . :   CUSTLIB     Primary group  . . . :   DPTAR
Object type. . . . :   *DTAARA     ASP device   . . . . :   *SYSBAS

Object secured by authorization list . . . . . :   *NONE

                        Object     -----------Object-----------
User        Group       Authority  Opr  Mgt  Exist  Alter  Ref
PGMR1                   *ALL        X    X     X      X     X
*GROUP      DPTAR       *CHANGE     X
DPTSM                   *USE        X
*PUBLIC                 *EXCLUDE
*ADOPTED                USER DEF         X     X      X     X
F3=Exit F11=Display data authorities F12=Cancel F17=Top F18=Bottom
```

If the user option (USROPT) field in USERB's user profile includes *EXPERT, this is how the display looks:

```
                          Display Object Authority

Object . . . . . :   CUSTNO     Owner  . . . . . :   PGMR1
  Library. . . . . :   CUSTLIB    Primary group  . . . :   DPTAR
Object type. . . . :   *DTAARA    ASP device  . . . . .:   *SYSBAS

Object secured by authorization list . . . . . :   *NONE

                 OBJECT     -----Object------   ------Data--------
User    Group    Authority   O   M   E   A   R   R   A   U   D   E
PGMR1            *ALL        X   X   X   X   X   X   X   X   X   X
*GROUP  DPTAR    *CHANGE     X                   X   X   X   X   X
DPTSM            *USE        X                   X               X
*PUBLIC          *EXCLUDE
*ADOPTED         USER DEF        X   X   X   X
```

Authority Reports

Several reports are available to help you monitor your security implementation. For example, you can monitor objects with *PUBLIC authority other than *EXCLUDE and objects with private authorities with the following commands:

- Print Public Authority (PRTPUBAUT)
- Print Private Authority (PRTPVTAUT)

For more information about security tools, see the *Tips and Tools for Securing Your iSeries*.

Working with Libraries

Two parameters on the Create Library (CRTLIB) command affect authority:

Authority (AUT): The AUT parameter can be used to specify either of the following:

- The public authority for the library
- 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       Owner  . . . . . :   PGMR1
  Library. . . . . :     QSYS        Primary group  . . . :   *NONE
Object type. . . . :   *LIB          ASP device   . . . . :   *SYSBAS

Object secured by authorization list. . . . . . . . . . . :   LIBLST

                         Object
User        Group        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 number . . . . . . . . . . . . . . . . . :   1
ASP device . . . . . . . . . . . . . . . . . :   *SYSBAS
Create authority . . . . . . . . . . . . . . :   *OBJLST
Create object auditing . . . . . . . . . . . :   *SYSVAL
Text description . . . . . . . . . . . . . . :   Customer Rec
```

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        Owner  . . . . . :   PGRM1
  Library. . . . . :   TESTLIB     Primary group  . . . :   *NONE
Object type. . . . :   *DTAARA     ASP device . . . . . :   *SYSBAS

Object secured by authorization list. . . . . :   OBJLST

                       Object
User        Group      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       Owner  . . . . . :   PGRM1
  Library  . . . . :   TESTLIB    Primary group  . . . :   *NONE
Object type. . . . :   *DTAARA    ASP device . . . . . :   *SYSBAS

Object secured by authorization list . . . . . :   *NONE

                       Object
User        Group      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. You can also use these commands to change object authority:

```
                          Edit Object Authority

Object. . . . . :   DTA1       Owner  . . . . . . . . :   PGMR1
  Library . . . . :   TESTLIB    Primary group  . . . :   *NONE
Object type.. . . :   *DTAARA    ASP device . . . . . :   *SYSBAS

Type changes to current authorities, press Enter.

  Object secured by authorization list  . . . . . . . . :   OBJLST

                        Object
User        Group       Authority
PGMR1                   *ALL
*PUBLIC                 *AUTL
```

Change Authority (CHGAUT)

Work with Authority (WRKAUT)

Grant Object Authority (GRTOBJAUT)

Revoke Object Authority (RVKOBJAUT)

To specify the generic authority subsets, such as Read/Write (*RX) or Write/Execute (*WX), you must use the CHGAUT or WRKAUT commands.

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   Owner  . . . . . :   PGMR1
  Library. . . . . :   TESTLIB     Primary group  . . . :   *NONE
Object type. . . . :   *FILE       ASP device . . . . . :   *SYSBAS

Type changes to current authorities, press Enter.

  Object secured by authorization list. . . . . :   LIST2

                       OBJECT          ----------Object-----------
User        Group      Authority  Opr  Mgt  Exist  Alter  Ref
PGMR1                  USER DEF    X    X            X     X
*PUBLIC                *AUTL
```

You can press F11 (Display data authorities) to view or change the data authorities:

```
                          Edit Object Authority

 Object . . . . . :   CONTRACTS   Owner  . . . . . :   PGMR1
  Library . . . . . :   TESTLIB     Primary group  . . . :   *NONE
 Object type. . . . :   *FIL        ASP device . . . . . :   *SYSBAS

Type changes to current authorities, press Enter.

 Object secured by authorization list. . . . . :   LIST2

                       OBJECT    --------------Data--------------
User        Group      Authority  Read  Add  Update  Delete  Execute
PGMR1                  USER DEF    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     Owner  . . . . . :   PGMR1
  Library. . . . . :   LIBB      Primary group  . . . :   *NONE
Object type. . . . :   *FILE     ASP device   . . . . :   *SYSBAS

  Object secured by authorization list. . . . . :   *NONE

                       Object     --------Object------------
User     Group   Authority  Read  Add  Update  Delete  Execute
USERB            USER DEF         X    X       X       X
```

```
                        Display Object Authority

Object . . . . . :   FILEB     Owner  . . . . . :   PGMR1
  Library. . . . . :   LIBB      Primary group  . . . :   *NONE
Object type  . . . :   *FILE     ASP device . . . . . :   *SYSBAS

tion list  . . . . . .   *NONE

                      Object    ---------------Data---------------
User        Group       Authority  Read  Add  Update  Delete  Execute
PGMR1                   USER DEF
```

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
ASP device  . . . . . .  *
Users  . . . . . .   *PUBLIC
+ for more values
Authority  . . . . . .  *USE
```

  This example for 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 GRTOBJAUT command with the REPLACE parameter.

```
GRTOBJAUT OBJ(TESTLIB/*ALL) OBJTYPE(*ALL) +
          USER(*PUBLIC) REPLACE(*YES)
```

  The REPLACE parameter indicates whether the authorities you specify replaces the existing authority for the user. The default value of REPLACE(*NO) gives the authority that you specify, but it does not remove any authority that is greater than the authority you specify, unless you are granting *EXCLUDE authority.

  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
ASP device  . . . . . .  *
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
ASP device  . . . . . .  *

  :
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
ASP device  . . . . . .  *
Users  . . . . . .   *PUBLIC
              + for more values
Authority  . . . . . .   *AUTL
```

Working with Object Ownership

To change ownership of an object, use one of the following:

The Change Object Owner (CHGOBJOWN) command

The Work with Objects by Owner (WRKOBJOWN) command

The Change Owner (CHGOWN) 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
                                                                  ASP
Opt  Object        Library       Type      Attribute          Device
     COPGMMSG      COPGMLIB      *MSGQ                        *SYSBAS
9    CUSTMAS       CUSTLIB       *FILE                        *SYSBAS
9    CUSTMSGQ      CUSTLIB       *MSGQ                        *SYSBAS
     ITEMMSGQ      ITEMLIB       *MSGQ                        *SYSBAS

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

You cannot delete a user profile that owns objects. The topic "Deleting User Profiles" on page 109 shows methods for handling owned objects when deleting a profile.

The Work with Objects by Owner display includes integrated file system objects. For these objects, the *Object* column on the display shows the first 18 characters of the path name. If the path name is longer than 18 characters, a greater than symbol (>) appears at the end of the path name. To see the absolute path name, place your cursor anywhere on the path name and press the F22 key.

Working with Primary Group Authority

To change the primary group or primary group's authority to an object, use one of the following commands:

Change Object Primary Group (CHGOBJPGP)

Work with Objects by Primary Group (WRKOBJPGP)

Change Primary Group (CHGPGP)

When you change an object's primary group, you specify what authority the new primary group has. You can also revoke the old primary group's authority. If you do not revoke the old primary group's authority, it becomes a private authority.

The new primary group cannot be the owner of the object.

To change an object's primary group, you must have all of the following:

- *OBJEXIST authority for the object.

- If the object is a file, library, or subsystem description, *OBJOPR and *OBJEXIST authority.
- If the object is an authorization list, *ALLOBJ special authority or be the owner of the authorization list.
- If revoking authority for the old primary group, *OBJMGT authority.
- If a value other than *PRIVATE is specified, *OBJMGT authority and all the authorities being given.

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 "Planning Authorization Lists" on page 227 for information about the advantages of using authorization lists.

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 114 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

You do not need any authority to the QSYS library to create an authorization list into that 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 127 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          Primary group  . . . :   *NONE

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): Each user's authority to the list is actually stored as a private authority in that

```
                              Add New Users

Object . . . . . :   CUSTLST1     Owner . . . PGMR1
  Library  . . . . . :    QSYS

Type new users, press Enter.

            Object    List
User        Authority  Mgt
AMES        *CHANGE
SMITHR      *CHANGE
```

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       Owner  . . . . . :   PGMR1
  Library  . . . . :   TESTLIB     Primary group. . . :   *NONE
Object type  . . . :   *FILE       ASP device . . . . :   *SYSBAS

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 authorization 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.

```
                    Display Authorization List Objects

Authorization list . . . . . :   CUSTLST1
  Library  . . . . . :   CUSTLIB
Owner  . . . . . :   OWNAR
Primary group  . . . . . :   DPTAR

                                                Primary
Object      Library     Type      Owner         group       Text
CUSTMAS     CUSTLIB     *FILE     OWNAR
CUSTADDR    CUSTLIB     *FILE     OWNAR
```

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.

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 library or directory path that contains the object. If the authority to the library or directory path 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 135 for more information about adopted authority.

The system verifies a user's authority to an object in the following order:

1. Object's authority - fast path
2. User's *ALLOBJ special authority
3. User's specific authority to the object
4. User's authority on the authorization list securing the object
5. Groups' *ALLOBJ special authority
6. Groups' authority to the object
7. Groups' authority on the authorization list securing the object
8. Public authority specified for the object or for the authorization list securing the object
9. Program owner's authority, if adopted authority is used

Note: Authority from one or more of the user's groups may be accumulated to find sufficient authority for the object being accessed.

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.

The numbers at the upper left of figures on the flowcharts are used in the examples following the flowcharts.

The steps representing the search of a profile's private authorities are highlighted:

Step 6 in Flowchart 3 on page 161

Step 6 in Flowchart 6 on page 167

Step 2 in Flowchart 8B on page 172

Repeating these steps is likely to cause performance problems in the authority checking process.

Flowchart 1: Main Authority Checking Process

The steps in Flowchart 1 show the main process the system follows in checking authority for an object.

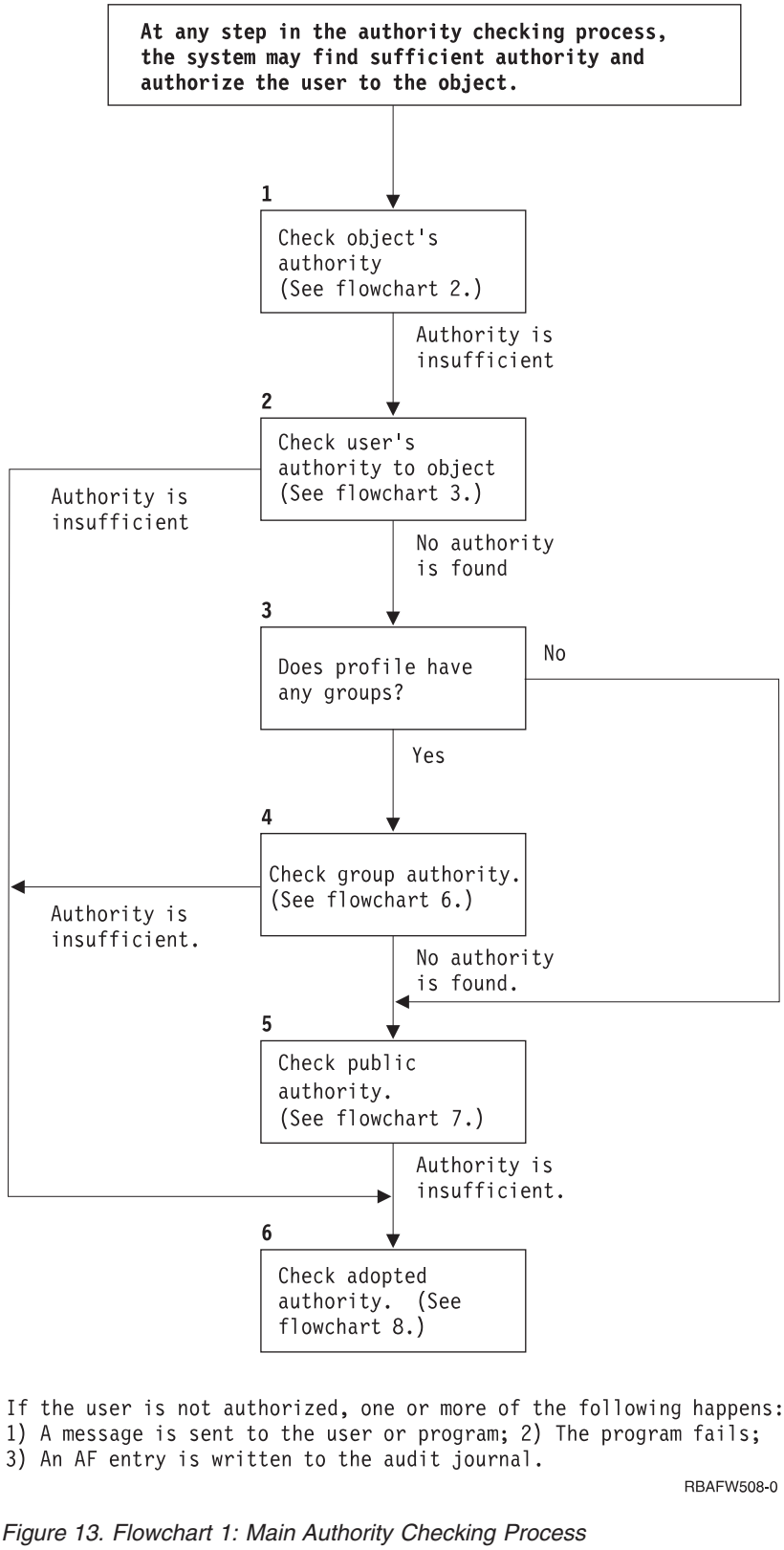


Figure 13. Flowchart 1: Main Authority Checking Process

Description of Flowchart 1: Main Authority Checking Process

Note: At any step in the authority checking process, the system may find sufficient authority and authorize the user to the object.

1. The system checks the object's authority. (Refer to Flowchart 2: Fast Path for Object Authority Checking.) If the system finds that authority is insufficient, it proceeds to Step 2.
2. The system checks the user's authority to the object. (Refer to Flowchart 3: How User Authority to an Object Is Checked.) If the system determines that the user does not have authority to the object, it proceeds to Step 3. If the system finds that the user's authority is insufficient, it proceed to Step 6.
3. The system checks whether the user profile belongs to any groups. If it does, the system proceeds to Step 4. If it does not, the system proceed to Step 5.
4. The system determines the group authority. (Refer to Flowchart 6). If the system determines that the group does not have authority to the object, it proceeds to Step 5. If the system determines that the group does not have sufficient authority to the object, it proceeds to Step 6.
5. The system checks the public authority of the object. (Refer to Flowchart 7.) If the system determines that the public authority is insufficient, it proceeds to Step 6.
6. The system checks the adopted authority of the object. (Refer to Flowchart 8.)

If the user is not authorized, one or more of the following happens:

- A message is sent to the user or program
- The program fails
- An AF entry is written to the audit journal

Flowchart 2: Fast Path for Object Authority Checking

The steps in Flowchart 2 are performed using information stored with the object. This is the fastest method for authorizing a user to an object.

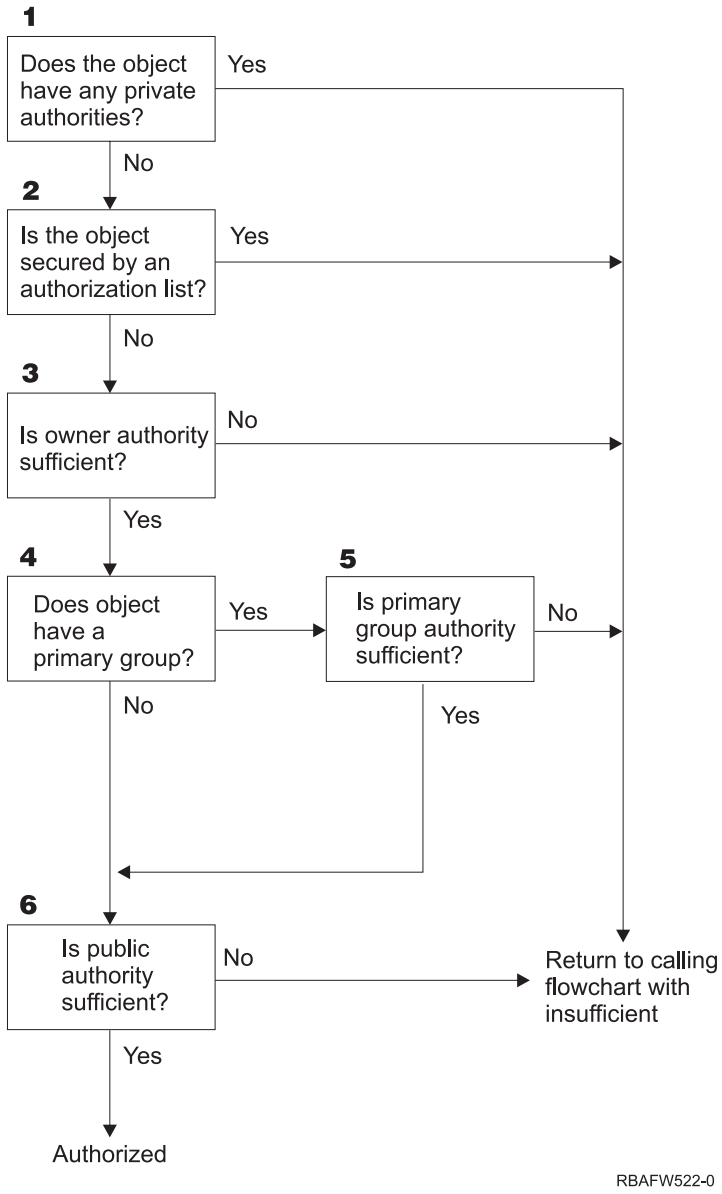


Figure 14. Flowchart 2: Fast Path for Object Authority

Description of Flowchart 2: Fast Path for Object Authority

1. The system determines whether the object has any private authorities. If it does, the system returns to calling flowchart with insufficient. If it does not, the system proceeds to Step 2.
2. The system determines whether the object is secured by an authorization list. If it is, the system returns to calling flowchart with insufficient. If it does not, the system proceeds to Step 3.
3. The system determines whether the owner of the object has sufficient authority. If it does, the system returns to calling flowchart with insufficient. If it does not, the system proceeds to Step 4.
4. The system determines whether the object has a primary group. If it does, the system proceeds to Step 5. If it does not the system proceeds to Step 6.
5. The system determines whether the object's primary group has sufficient authority. If it does, the system proceeds to Step 6. If it does not, the system returns to calling flowchart with insufficient.

6. The system determines whether public authority is sufficient. If it is, the object is authorized. If it is not, the system returns to calling flowchart with insufficient.

Flowchart 3: How User Authority to an Object Is Checked

The steps in Flowchart 3 are performed for the individual user profile.

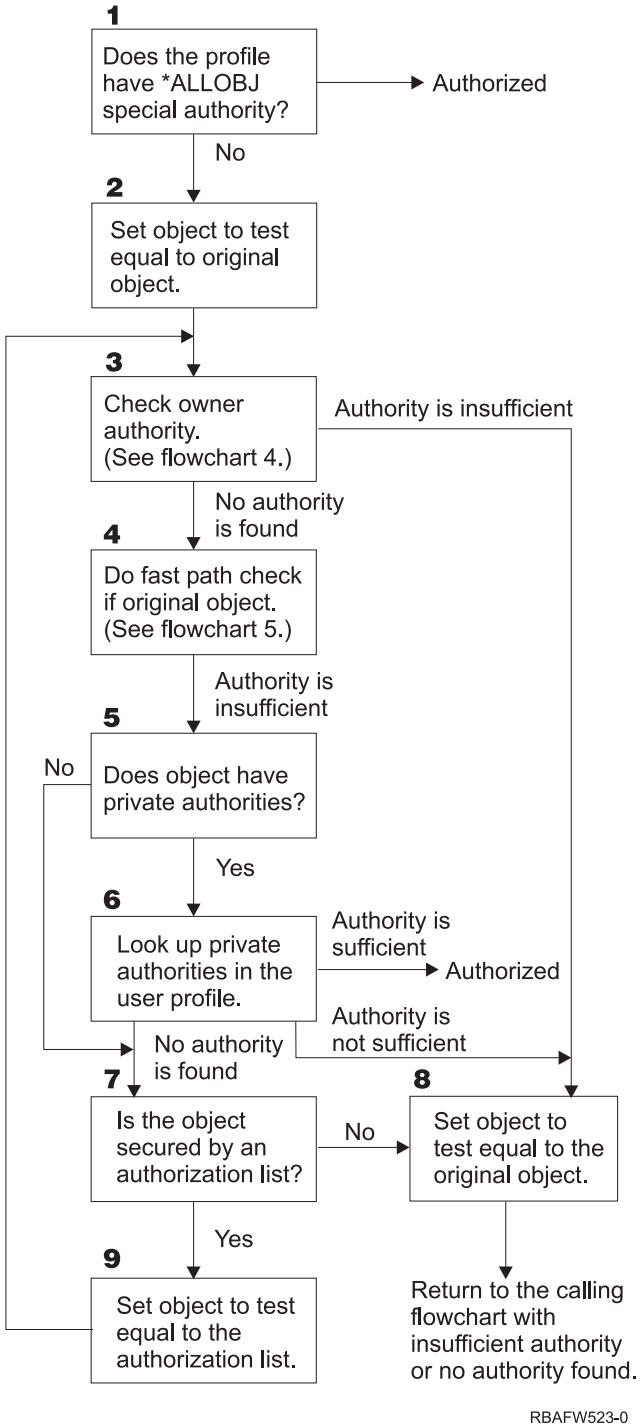


Figure 15. Flowchart 3: Check User Authority

Description of Flowchart 3: Check User Authority

1. The system determines if the user profile has *ALLOBJ authority. If the profile does have *ALLOBJ authority, then the profile is authorized. If it does not have *ALLOBJ authority, then the authority checking proceeds to Step 2.
2. The system sets the authority of the object to the equal the original object. The authority checking proceeds to Step 3.
3. The system check the owner authority. If the authority is unsufficient, then it proceeds to Step 8. If no authority is found, then it proceeds to Step 4.
4. The system completes a fast path authority check of the original object. (Refer to Flowchart 5). If authority is insufficient, then authority checking proceeds to Step 5.
5. The system determines if the object has private authorities. If it does, then the authority check proceeds to Step 6. If there are no private authorities, then the authority checking goes to Step 7.
6. The system check for private authorities with the user profile. If the authority is sufficient, then the user is authorized. If authority is not sufficient, then the authority checking proceeds to Step 8. If no authority is found, then the authority checking proceeds to Step 7.
7. The system determines if the object is secured by an authorization list. If it is not, then the authority checking proceeds to Step 8. If it is secured by an authorization list, then the authority checking proceeds to Step 9.
8. The system sets the object to test equal to the original object and returns to the calling flowchart with insufficient authority or no authority found.
9. The system sets the object to test equal to the authorization list and returns to Step 3.

Flowchart 4: How Owner Authority Is Checked

Figure 16 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 the owner's 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.

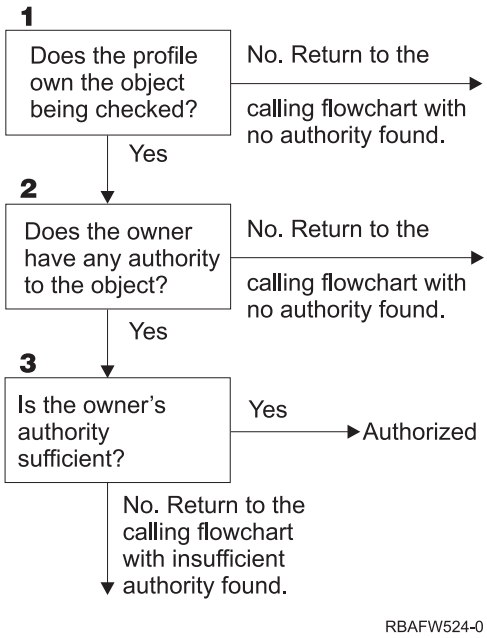


Figure 16. Flowchart 4: Owner Authority Checking

Description of Flowchart 4: Owner Authority Checking

1. The system determines if the user profile owns the object being checked. If the user profile does own the object, then in moves to Step 2. If the user profile does not own the object, then the system returns to the calling flowchart with no authority found.
2. If the user profile does own the object, the system then determines if the owner has authority to the object. If he or she is the owner, then the authority check proceeds to Step 3. If the system determines that the owner does not have authority to the object, then the system returns to the calling flowchart with no authority found.
3. If the owner does have authority to the object, then the system determines whether or not this authority is sufficient to access to object. If the authority is sufficient, then the owner is authorized to the object. If it is not sufficient, then the system returns to the calling flowchart with insufficient authority found.

Flowchart 5: Fast Path for User Authority Checking

Figure 17 on page 164 shows the fast path for testing user authority without searching private authorities.

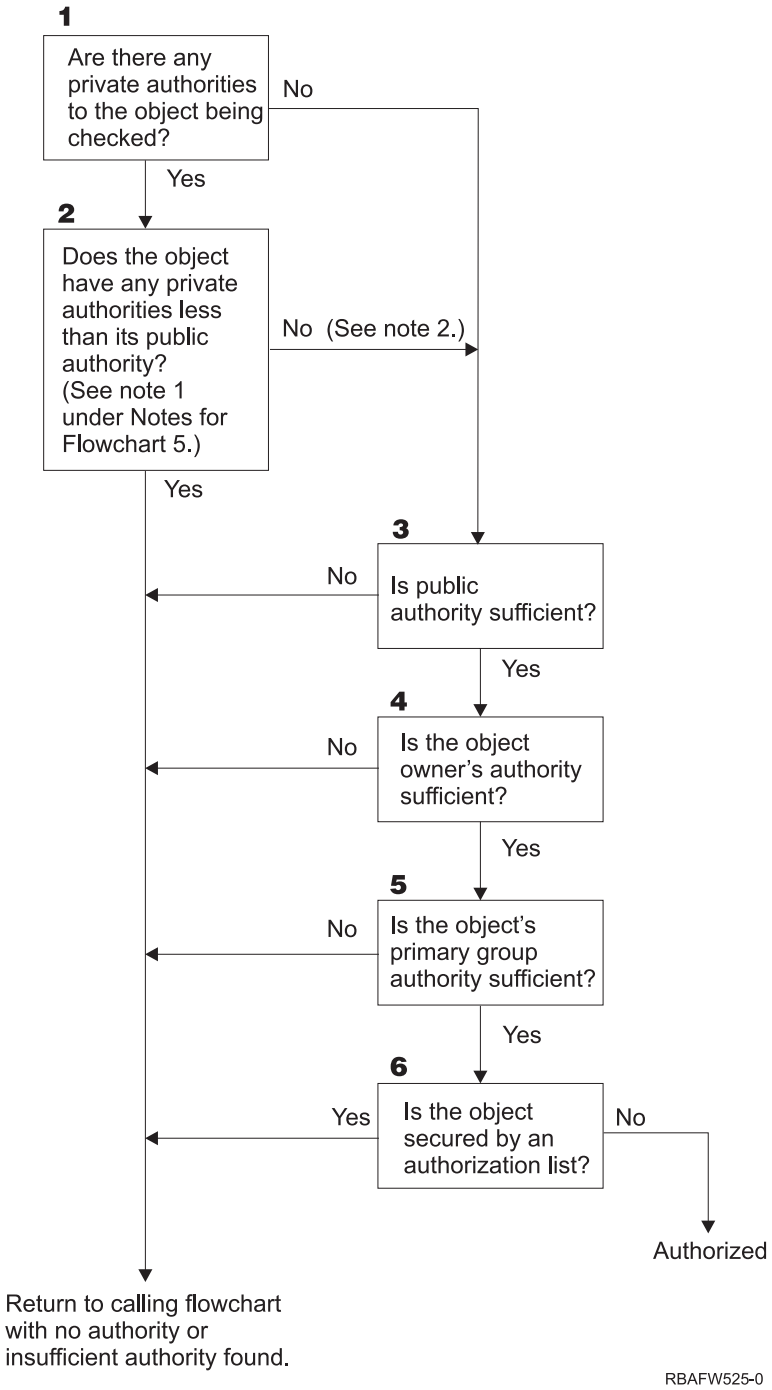


Figure 17. Flowchart 5: Fast Path for User Authority

Notes for Flowchart 5:

1. Authority is considered less than public if any authority that is present for *PUBLIC is not present for another user. In the example shown in Table 106, the public has *OBJOPR, *READ, and *EXECUTE authority to the object. WILSONJ has *EXCLUDE authority and does not have any of the authorities the public has. Therefore, this object does have private authority less than its public authority. (OWNAR also has less authority than the public, but owner authority is not considered private authority.)

Table 106. Public versus Private Authority

	Users			
Authority	**OWNAR**	**DPTMG**	**WILSONJ**	***PUBLIC**
Object Authorities:				
*OBJOPR		X		X
*OBJMGT	X			
*OBJEXIST				
*OBJALTER				
*OBJREF				
Data Authorities				
*READ		X		X
*ADD		X		
*UPD		X		
*DLT		X		
*EXECUTE		X		X
*EXCLUDE			X	

2. This path provides 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.

Description of Flowchart 5: Fast Path for User Authority

This flowchart shows the fast path for testing user authority without searching private authorities.

1. The system determines if there are any private authorities to the object being checked. If there are private authorities to the object then the authority check proceeds to Step 2. If there is no private authority, the authority check proceeds to Step 3.
2. If private authorities exist, then the system determines if the object has private authorities that are less than its public authority. (See note 1.) If the object does have private authorities that are less than its public authority, then the system returns to the calling flowchart with no authority or insufficient authority found. If the object does not have private authorities that are less than its public authority, (See note 2), then the authority check proceeds to Step 3.
3. If the object does not have private authorities that are less than its public authority, then the system determine if the public authority is sufficient. If the public authority is sufficient, then the authority check proceeds to Step 4. If the public authority is insufficient, then system returns to the calling flowchart with no authority or insufficient authority found.
4. If the public authority is sufficient, then the system determines if the object owner's authority is sufficient. If the object owner's authority is sufficient, then the authority check proceeds to Step 5. If the object owner's authority is insufficient, then system returns to the calling flowchart with no authority or insufficient authority found.
5. If the object owner's authority is sufficient, then the system determines if the object's primary group authority is sufficient. If the object's primary group authority is sufficient, then the authority check proceeds to Step 6. If object's primary group authority is insufficient, then the system returns to the calling flowchart with no authority or insufficient authority found.

6. If the object's primary group authority is sufficient, then the system determines if the object is secured by an authorization list. If the object is secured by an authorization list, then the system returns to the calling flowchart with no authority or insufficient authority found. If the object is not secured by an authorization list, then the user is authorized to the object.

Flowchart 6: How Group Authority Is Checked

A user may be a member of up to 16 groups. A group may have private authority to an object, or it may be the primary group for an object.

Authority from one or more of the user's groups may be accumulated to find sufficient authority for the object being accessed. For example, WAGNERB needs *CHANGE authority to the CRLIM file. *CHANGE authority includes *OBJOPR, *READ, *ADD, *UPD, *DLT, and *EXECUTE. Table 107 shows the authorities for the CRLIM file:

Table 107. Accumulated Group Authority

	Users			
Authority	**OWNAR**	**DPT506**	**DPT702**	***PUBLIC**
Object Authorities:				
*OBJOPR	X	X	X	
*OBJMGT	X			
*OBJEXIST	X			
*OBJALTER	X			
*OBJREF	X			
Data Authorities				
*READ	X	X	X	
*ADD	X	X		
*UPD	X	X	X	
*DLT	X		X	
*EXECUTE	X	X	X	
*EXCLUDE				X

WAGNERB needs both DPT506 and DPT702 to get sufficient authority to the CRLIM file. DPT506 is missing *DLT authority, and DPT702 is missing *ADD authority.

Flowchart 6 on page 167 shows the steps in checking group authority.

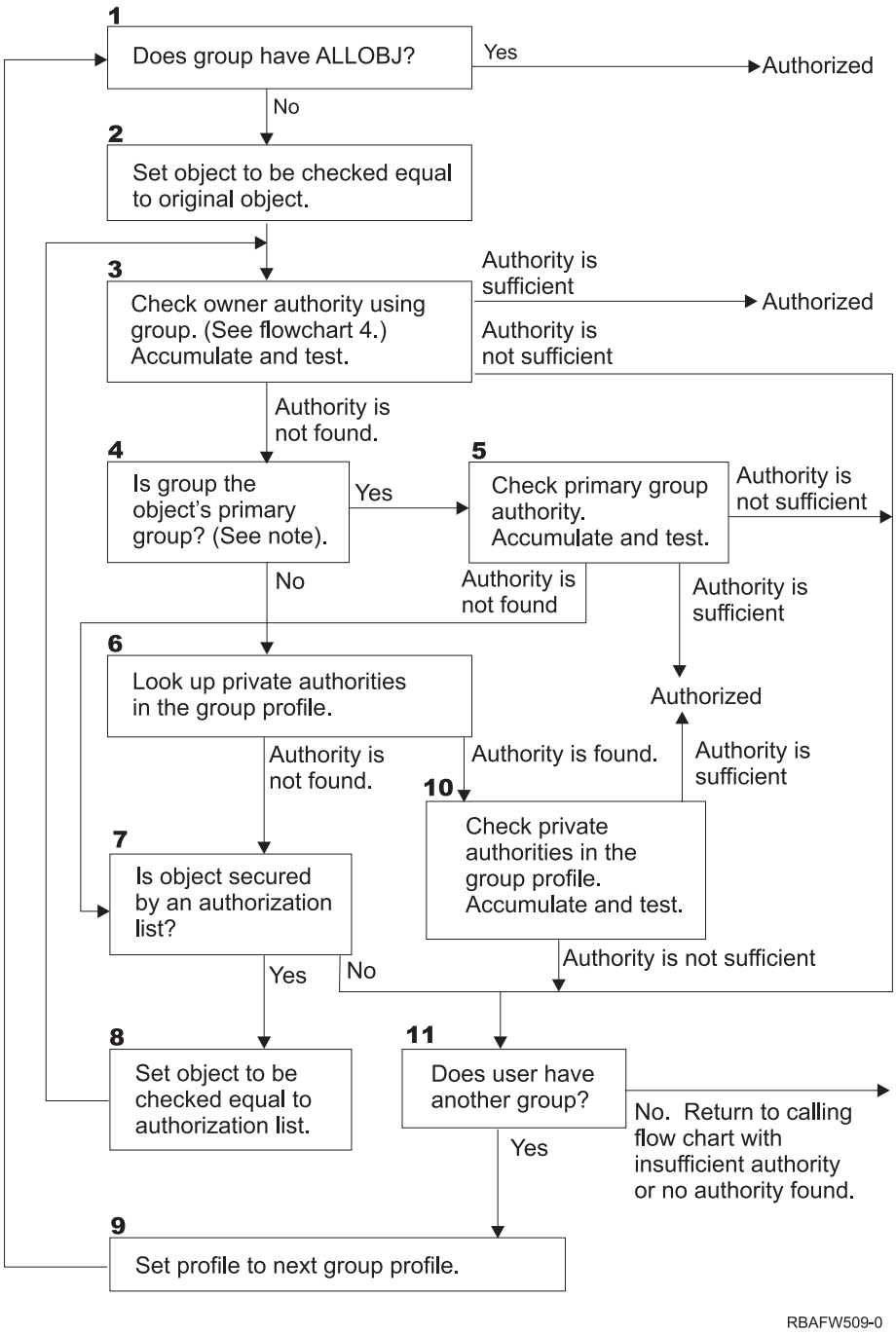


Figure 18. Flowchart 6: Group Authority Checking

Note: If the user is signed on as the profile that is the primary group for an object, the user cannot receive authority to the object through the primary group.

Description of Flowchart 6: Group Authority Checking

1. The system determines if the group has ALLOBJ authority. If it does, then the group is authorized. If it does not, authority checking proceeds to Step 2.
2. If the group does not have ALLOBJ authority, the system sets the object that is being checked to be equal to the original object.
3. After the system sets the object to the original, it checks owner authority (See Flowchart 4) If authority is sufficient, then the group is authorized. If the

authority is not sufficient, then the authority check goes to Step 7. If the authority is not found, then the authority check proceeds to Step 4.

4. If the owner authority is not found, then the system checks if the group is the object's primary group.

   Note: If the user is signed on as the profile that is the primary group for an object, the user cannot receive authority to the object through the primary group.

   If the group is the object's primary group, then the authority check proceeds to Step 5. If the group is not the object's primary group, then authority check proceeds to Step 6.
5. If the group is the object's primary group, then the system checks and tests the primary group authority. If primary group authority is sufficient, then the group is authorized. If primary group authority is insufficient or is not found, then the authority check goes to Step 7.
6. If the group is not the object's primary group, then the system looks up the private authorities in the group profile. If authority is found then authority checking goes to Step 10. If authority is not found then authority checking proceeds to Step 7.
7. If no authority is found for the private authorities for the group profile then the system checks to see if the object is secured by an authorization list. If the object is secured by an authorization list, then the authority check proceeds to Step 8. If the object is not secured by an authorization list then the authority check goes to Step 11.
8. If the object is secured by an authorization list, then the system set the object to be checked equal to the authorization list and authority check returns to Step 3.
9. If the user does belong to another group profile, then the system sets this profile to the next group profile and returns to Step 1 to start the authority checking process over again.
10. If authority is found for private authorities within the group profile, then the private authorities are checked and tested in the group profile. If authorities are sufficient, then the group profile is authorized. If it is not sufficient then the authority check goes to Step 7.
11. If an object is not secured by an authorization list, then the system checks to see if the users is associated with another group profile. If the user does belong to another group profile, then the system goes to Step 9. If the user does not belong to another group profile then the system returns to the calling flowchart with insufficient authority or no authority found.

Flowchart 7: How Public Authority Is Checked

When checking public authority, the system must determine whether to use the public authority for the object or the authorization list. Flowchart 7 shows the process:

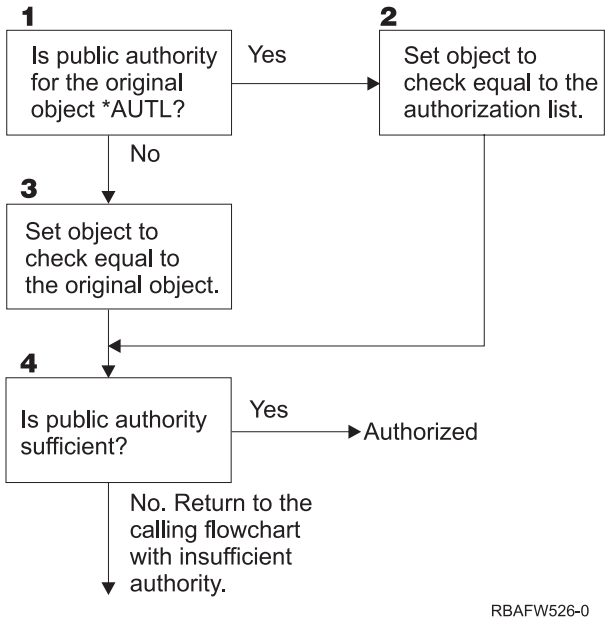


Figure 19. Flowchart 7: Check Public Authority

Description of Flowchart 7: Check Public Authority

Flowchart 7 shows how the system must determine whether to use the public authority for the object or the authorization list.

1. The system determine if the public authority for the original object is *AUTL. If the public authority for the original object is *AUTL, then the system proceeds to Step 2. If the public authority for the original object is not *AUTL, then the system proceeds to Step 3.
2. If the public authority for the original object is *AUTL, then the system sets the object being checked equal to the authorization list and proceeds to Step 4.
3. If the public authority for the original object is not *AUTL, then the system sets the object being checked to the original object and proceeds to Step 4.
4. If the object being checked has been set equal to the authorization list or the original object, the system determines of the public authority is sufficient. If the public authority is sufficient then user is authorized to the object. If the public authority is not sufficient then the system returns to the calling flowchart with insufficient authority.

Flowchart 8: 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 checks to see if the program owner has *ALLOBJ special authority or owns the object being tested. This is repeated for every program in the stack that uses adopted authority.

If sufficient authority is not found, the system checks to see if the program owner has private authority for the object being checked. This is repeated for every program in the stack that uses adopted authority.

Figure 20 and Figure 21 on page 172 show the process for checking adopted authority.

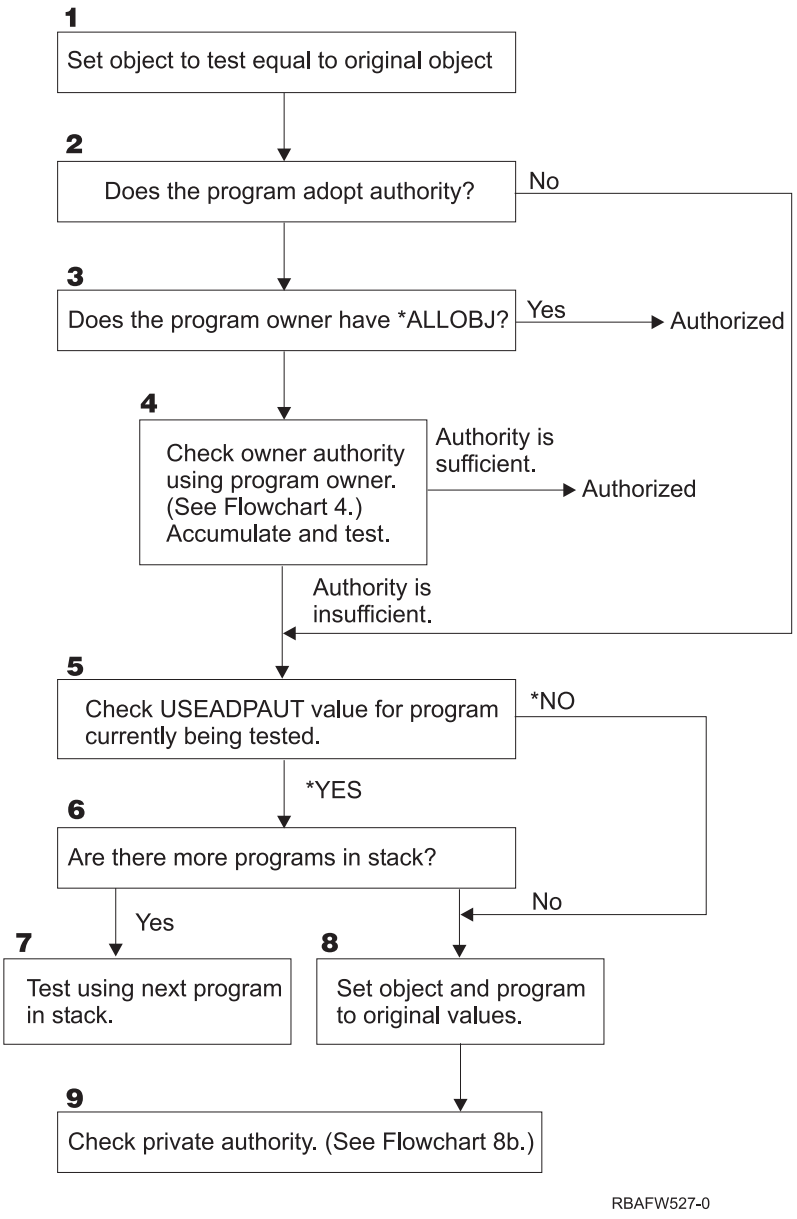


*Figure 20. Flowchart 8A: Checking Adopted Authority User *ALLOBJ and Owner*

Description of Flowchart 8A: Checking Adopted Authority User *ALLOBJ and Owner

Flowchart 8A describes how the system checks adopted authority when insufficient authority has been found by checking user authority.

1. The system sets the object being checked to the original object and proceeds to Step 2.
2. The system determines if the program adopts authority. If the program does adopt authority then the authority checking proceeds to Step 3. If the program does not adopt authority and the authority is insufficient, then authority checking goes to Step 5.

3. If the program does adopt authority, then the system determines if the program owner has *ALLOBJ authority. If the program owner does have *ALLOBJ authority, then the user is authorized. If the program owner does not have *ALLOBJ authority, then the authority checking proceeds to Step 4.
4. If the program owner does not have *ALLOBJ authority, then the system checks and tests the owner authority. If the authority is sufficient, then the user is authorized. If the authority is insufficient then authority checking proceeds to Step 5.
5. The system checks USEADPAUT value for the program currently being test. If the value equals *NO then authority checking proceeds to Step 8. If the value is equal to *YES then the authority checking proceeds to Step 6.
6. If the USEADPAUT value is equal to *YES, then the system determine if there are more programs waiting in the stack. If there are more programs in the stack, then authority checking proceeds to Step 7. If there are not any more programs waiting in the stack, then authority checking goes to Step 8.
7. If there are more programs in the stack, the system test the next program in the stack.
8. If there are no more programs in the stack or the USEADPAUT value is equal to *NO, then system sets the object and program to the original values and proceeds to Step 9.
9. The system check private authority. This is described in Flowchart 8B: Checking Adopted Authority Using Private Authorities.

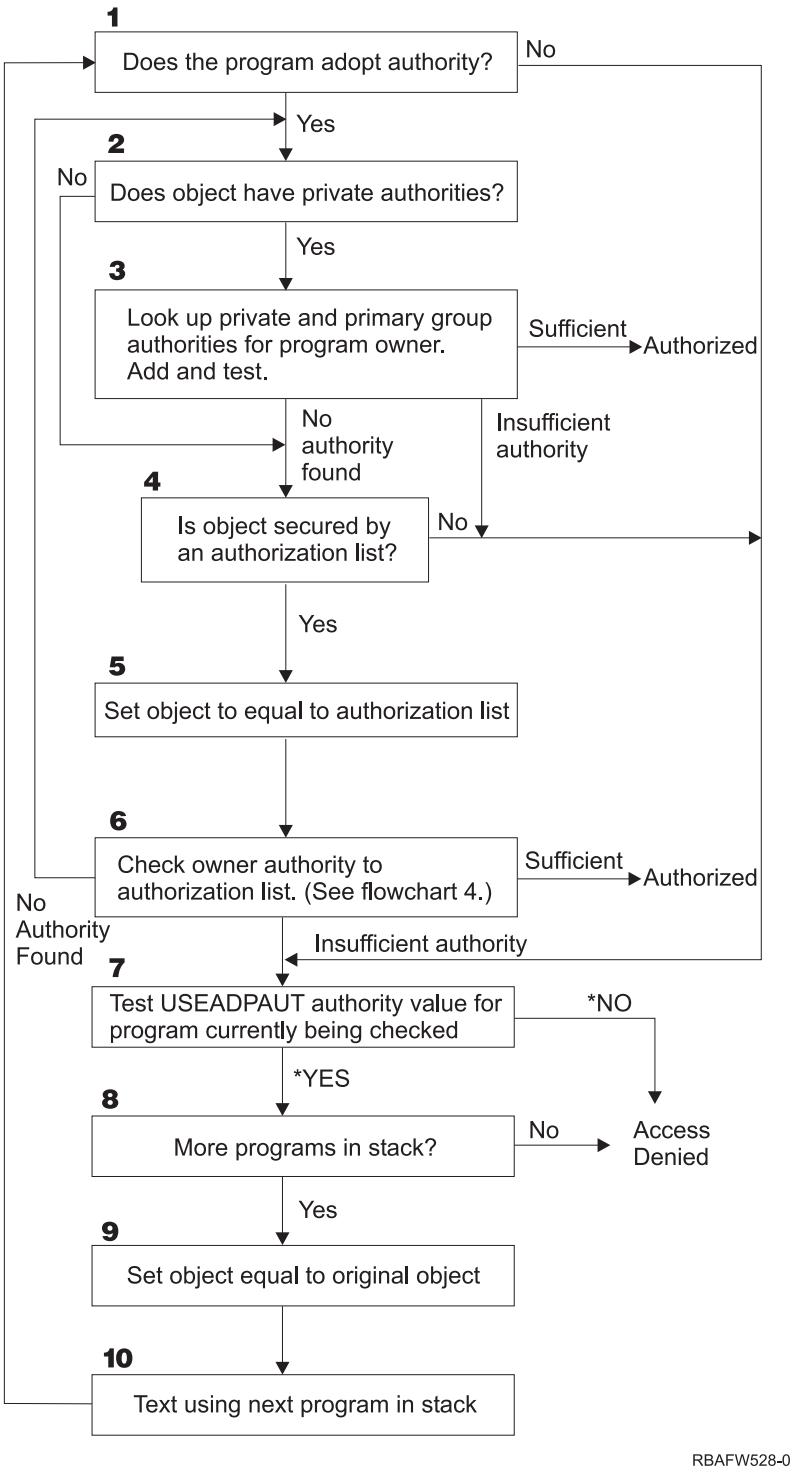


Figure 21. Flowchart 8B: Checking Adopted Authority Using Private Authorities

Description of Flowchart 8B: Checking Adopted Authority Using Private Authorities

1. The system determines whether the program can adopt authority. If yes, proceed to Step 2. If no, proceed to Step 7.
2. The system determines whether the object has private authorities. If yes, proceed to Step 3. If no, proceed to Step 4.

3. The system checks the private and primary group authorities for the program owner. If authority is sufficient, the program is authorized. If insufficient authority is found, proceed to Step 7. If no authority is found, proceed to Step 4.
4. The system determines whether the object is secured by an authorization list. If yes, proceed to Step 5. If no, proceed to Step 7.
5. The system sets object equal to authorization list and then proceeds to Step 6.
6. The system checks the owner's authority to the authorization list. (Refer to Flowchart 4.) If not authority is found, go back to Step 2. If sufficient authority is found, the program is authorized.
7. The system tests the USEADPAUT authority value for the program currently being checked. If *YES, proceed to Step 8. If *NO, access denied.
8. The system checks whether there are more programs in the stack. If yes, proceed to Step 9. If no, access denied.
9. The system sets object equal to original object and proceeds to Step 10.
10. Text using next program in stack and start back at Step 1.

Authority Checking Examples

Following are several examples of authority checking. These examples demonstrate the steps the system uses to determine whether a user is allowed a requested access to an object. These examples are intended to show how authority checking works and where potential performance problems may occur.

Figure 22 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 6) 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          Owner  . . . . . :   OWNCP
  Library  . . . . :   CONTRACTS     Primary group  . . :   *NONE
Object type    . . :   *FILE          ASP device . . . . :   *SYSBAS

Object secured by authorization list  . . . . . . . . . . . :   *NONE

                        Object
User        Group       Authority
OWNCP                   *ALL
DPTSM                   *CHANGE
DPTMG                   *CHANGE
WILSONJ                 *USE
*PUBLIC                 *USE
```

Figure 22. Authority for the PRICES File

Case 1: Using Private 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.
   a. Flowchart 2, step 1.
2. Flowchart 1, step 2.
   a. Flowchart 3, steps 1 and 2. Object to check = CONTRACTS/PRICES *FILE.
   b. Flowchart 3, step 3.
      1) Flowchart 4, step 1. Return to Flowchart 3 with no authority found. ROSSM does not own the PRICES file.
   c. Flowchart 3, step 4.
      1) Flowchart 5, steps 1, 2, and 3. Public is not sufficient.
   d. Flowchart 3, step 5.
   e. **Flowchart 3, step 6**. ROSSM does not have private authority to the PRICES file.
   f. Flowchart 3, steps 7 and 8. The PRICES file is not secured by an authorization list. Return to Flowchart 1 with no authority found.
3. Flowchart 1, steps 3 and 4. DPTSM is the group profile for ROSSM.
   a. Flowchart 6, steps 1, 2, and 3.
      1) Flowchart 4, step 1. DPTSM does not own the PRICES file.
   b. Flowchart 6, step 4. DPTSM is not the primary group for the PRICES file.
   c. **Flowchart 6, step 6**. Authorized. (DPTSM has *CHANGE authority.)

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 private group authority usually causes two searches of private authorities (for the user and the group), when public authority is not adequate. One search of the private authority could have been avoided by making DPTSM the primary group for the PRICES file.

Case 2: Using Primary Group Authority

ANDERSJ needs *CHANGE authority to the CREDIT file. ANDERSJ is a member of the DPTAR group. Neither ANDERSJ nor DPTAR has *ALLOBJ special authority. Figure 23 shows the authorities for the CREDIT file.

```
                          Display Object Authority

Object . . . . . :   CREDIT          Owner  . . . . . :   OWNAR
  Library  . . . . :   ACCTSRCV      Primary group  . . :   DPTAR
Object type    . . :   *FILE         ASP device . . . . :   *SYSBAS

Object secured by authorization list  . . . . . :   *NONE

                       Object
User        Group      Authority
OWNAR                  *ALL
DPTAR                  *CHANGE
*PUBLIC                *USE
```

Figure 23. Authority for the CREDIT File

The system performs these steps to determine whether to allow ANDERSJ to have *CHANGE access to the CREDIT file:

1. Flowchart 1, step 1.
   a. Flowchart 2, step 1. DPTAR's authority is primary group authority, not private authority.
   b. Flowchart 2, steps 2, 3, 4, 5, and 6. Public authority is not sufficient.
2. Flowchart 1, step 2.
   a. Flowchart 3, steps 1 and 2. Object to check = ACCTSRCV/CREDIT *FILE.
   b. Flowchart 3, step 3.
      1) Flowchart 4, step 1. ANDERSJ does not own the CREDIT file. Return to Flowchart 3 with no authority found.
   c. Flowchart 3, step 4.
      1) Flowchart 5, step 1. The CREDIT file has no private authorities.
      2) Flowchart 5, step 3. Public authority is not sufficient. Return to Flowchart 3 with no authority found.
   d. Flowchart 3, steps 5, 7, and 8. The CREDIT file is not secured by an authorization list. Return to Flowchart 1 with no authority found.
3. Flowchart 1, steps 3 and 4. ANDERSJ is a member of the DPTAR group profile.
   a. Flowchart 6, steps 1 and 2. Object to check = ACCTSRCV/CREDIT *FILE.
   b. Flowchart 6, step 3.
      1) Flowchart 4, step 1. DPTAR does not own the CREDIT file. Return to Flowchart 6 with no authority found.
   c. Flowchart 6, steps 4 and 5. Authorized. DPTAR is the primary group for the CREDIT file and has *CHANGE authority.

Result: ANDERSJ is authorized because DPTAR is the primary group for the CREDIT file and has *CHANGE authority.

Analysis: If you use primary group authority, the authority checking performance is better than if you specify private authority for the group. This example does not require any search of private authorities.

Case 3: Using Public Authority

User JONESP wants to access the CREDIT 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 CREDIT file:

1. Flowchart 1, step 1.
   a. Flowchart 2, step 1. The CREDIT file has no private authorities. DPTAR's authority is primary group authority, not private authority.
   b. Flowchart 2, steps 2 and 3. Owner's authority (OWNAR) is sufficient.
   c. Flowchart 2, steps 4 and 5. Primary group authority (DPTAR) is sufficient.
   d. Flowchart 2, step 6. Authorized. Public authority is sufficient.

Analysis: This example shows the performance benefit gained when you avoid defining any private authorities for an object.

Case 4: Using Public Authority Without Searching Private 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.
   a. Flowchart 2, step 1. The PRICES file has private authorities.
2. Flowchart 1, step 2.
   a. Flowchart 3, steps 1 and 2. Object to check = CONTRACTS/PRICES *FILE.
   b. Flowchart 3, step 3.
      1) Flowchart 4, step 1. JONESP does not own the PRICES file. Return to Flowchart 3 with no authority found.
   c. Flowchart 3, step 4.
      1) Flowchart 5, steps 1, 2, and 3. Public authority is sufficient.
      2) Flowchart 5, step 4. Owner authority is sufficient. (OWNCP has *ALL.)
      3) Flowchart 5, step 5. The PRICES file does not have a primary group.
      4) Flowchart 5, step 6. Authorized. (The PRICES file is not secured by an authorization list.)

Analysis: This example shows the performance benefit gained when you avoid defining any private authorities for an object that are less than public authority. 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 5: 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.
   a. Flowchart 2, step 1.
2. Flowchart 1, step 2.
   a. Flowchart 3, steps 1 and 2. Object to check = CONTRACTS/PRICES *FILE.
   b. Flowchart 3, step 3.
      1) Flowchart 4, step 1. SMITHG does not own the PRICES file. Return to Flowchart 3 with no authority found.
   c. Flowchart 3, step 4.
      1) Flowchart 5, steps 1, 2, and 3. Public is not sufficient.
   d. Flowchart 3, step 5.
   e. **Flowchart 3, step 6**. SMITHG does not have private authority.
   f. Flowchart 3, steps 7 and 8. The PRICES file is not secured by an authorization list. Return to Flowchart 1 with no authority found.
3. Flowchart 1, step 3. SMITHG does not have a group.
4. Flowchart 1, step 5.
   a. Flowchart 7, step 1. Public authority is not *AUTL.
   b. Flowchart 7, step 3. Object to check = CONTRACTS/PRICES *FILE.
   c. Flowchart 7, step 4. Public authority is not sufficient.
5. Flowchart 1, step 6.
   a. Flowchart 8A, step 1. Object to check = CONTRACTS/PRICES *FILE.
   b. Flowchart 8A, steps 2 and 3. OWNCP does not have *ALLOBJ authority.
   c. Flowchart 8A, step 4.

1) Flowchart 4, steps 1, 2, and 3. Authorized. OWNCP owns the PRICES files and has sufficient authority.

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 "Case 1: Using Private Group Authority" on page 173.

- 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.
- You could also change Case 1 so that DPTSM is the primary group for the PRICES file and has *CHANGE authority to it. If DPTSM is the first group for SMITHG (specified in the GRPPRF parameter of SMITHG's user profile), the performance characteristics would be the same as Case 5.

Case 6: 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.
   a. Flowchart 2, step 1. PRICES has private authorities.
2. Flowchart 1, step 2.
   a. Flowchart 3, steps 1 and 2. Object to check = CONTRACTS/PRICES *FILE.
   b. Flowchart 3, step 3.
      1) Flowchart 4, step 1. WILSONJ does not own the PRICES file. Return to Flowchart 3 with no authority found.
   c. Flowchart 3, step 4.
      1) Flowchart 5, steps 1, 2, and 3. Public is not sufficient.
   d. Flowchart 3, step 5.
   e. **Flowchart 3, step 6**. WILSONJ has *USE authority, which is not sufficient.
   f. Flowchart 3, step 8. Object to test = CONTRACTS/PRICES *FILE. Return to Flowchart 1 with insufficient authority.
3. Flowchart 1, step 6.
   a. Flowchart 8A, step 1. Object to check = CONTRACTS/PRICES *FILE.
   b. Flowchart 8A, step 2. Program CPPGM01 does not adopt authority.
   c. Flowchart 8A, step 5. The *USEADPAUT parameter for the CPPGM01 program is *NO.
   d. Flowchart 8A, steps 8 and 9.
      1) Flowchart 8B, step 1. Program CPPGM01 does not adopt authority.

2) Flowchart 8B, step 7. The *USEADPAUT parameter for the CPPGM01 program is *NO. Access is denied.

Analysis: This example demonstrates that a user can be denied access to an object even though the user's group has sufficient authority.

Giving a user the same authority as the public but less than the user's group does not affect the performance of authority checking for other users. However, if WILSONJ had *EXCLUDE authority (less than public), you would lose the performance benefits shown in Case 4.

Although this example has many steps, private authorities are searched only once. This should provide acceptable performance.

Case 7: Public Authority without Private Authority

The authority information for the ITEM file looks like this:

```
                          Display Object Authority

Object . . . . . :   ITEM            Owner  . . . . . :   OWNIC
  Library  . . . . :   ITEMLIB         Primary group  . . :   *NONE
Object type  . . . . :   *FILE           ASP device   . . . . :   *SYSBAS

Object secured by authorization list  . . . . . :   *NONE

                        Object
User        Group       Authority
OWNIC                   *ALL
*PUBLIC                 *USE
```

Figure 24. Display Object Authority

ROSSM needs *USE authority to the ITEM file. ROSSM is a member of the DPTSM group profile. These are the authority-checking steps:

1. Flowchart 1, step 1.
   a. Flowchart 2, steps 1, 2, and 3. OWNIC's authority is sufficient.
   b. Flowchart 2, step 4. The ITEM file does not have a primary group.
   c. Flowchart 2, step 6. Authorized. Public authority is sufficient.

Analysis: Public authority provides the best performance when it is used without any private authorities. In this example, private authorities are never searched.

Case 8: 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.
   a. Flowchart 2, steps 1, 2, 3, 4, and 6. Public authority is not sufficient.
2. Flowchart 1, step 2.
   a. Flowchart 3, steps 1 and 2. Object to check = ITEMLIB/ITEM *FILE.
   b. Flowchart 3, step 3.

1) Flowchart 4, step 1. WILSONJ does not own the ITEM file. Return to Flowchart 3 with no authority found.

c. Flowchart 3, step 4.

1) Flowchart 5, steps 1 and 3. Public authority is not sufficient. Return to Flowchart 3 with no authority found.

d. Flowchart 3, steps 5, 7, and 8. The ITEM file is not secured by an authorization list. Return to Flowchart 1 with no authority found.

3. Flowchart 1, steps 3 and 5. (WILSONJ does not have a group profile.)

a. Flowchart 7, steps 1, 3, and 4. The public has *USE authority, which is not sufficient.

4. Flowchart 1, step 6.

a. Flowchart 8A, step 1. Object to check = ITEMLIB/ITEM *FILE.

b. Flowchart 8A, steps 2, 3, and 4. The OWNIC profile does not have *ALLOBJ authority.

1) Flowchart 4, steps 1, 2, and 3. Authorized. OWNIC has sufficient authority to the ITEM file.

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 9: Using an Authorization List

The ARWKR01 file in library CUSTLIB is secured by the ARLST1 authorization list. Figure 25 and Figure 26 on page 180 show the authorities:

```
                          Display Object Authority

Object . . . . . :   ARWRK01       Owner  . . . . . :   OWNAR
  Library  . . . . :   CUSTLIB       Primary group  . . . :   *NONE
Object type  . . . . :   *FILE         ASP device . . . . . :   *SYSBAS

Object secured by authorization list. . . . . :   ARLST1

                        Object
User        Group       Authority
OWNCP                   *ALL
*PUBLIC                 *USE
```

Figure 25. Authority for the ARWRK01 File

```
                        Display Authorization List

Object . . . . . :   ARLST1      Owner  . . . . . :   OWNAR
  Library  . . . . . :   QSYS        Primary group  . . . :   *NONE

                       Object     List
User        Group      Authority  Mgt
OWNCP                  *ALL
AMESJ                  *CHANGE
*PUBLIC                *USE
```

Figure 26. 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.
   a. Flowchart 2, steps 1 and 2. The ARWRK01 file is secured by an authorization list.
2. Flowchart 1, step 2.
   a. Flowchart 3, steps 1 and 2. Object to check = CUSTLIB/ARWRK01 *FILE.
   b. Flowchart 3, step 3.
      1) Flowchart 4, step 1. AMESJ does not own the ARWRK01 file. Return to Flowchart 2 with no authority found.
   c. Flowchart 3, step 4.
      1) Flowchart 5, steps 1 and 3. Public authority is not sufficient. Return to Flowchart 3 with no authority found.
   d. Flowchart 3, steps 5, 7, and 9. Object to check = ARLST1 *AUTL.
   e. Flowchart 3, step 3.
      1) Flowchart 4, step 1. AMESJ does not own the ARLST1 authorization list. Return to Flowchart 3 with no authority found.
   f. Flowchart 3, steps 4 and 5.
   g. **Flowchart 3, step 6**. Authorized. AMESJ has *CHANGE authority to the ARLST1 authorization list.

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 private authorities, authorization lists, and group profiles are combined, as in "Case 11: Combining Authorization Methods" on page 181.

Case 10: Using Multiple Groups

WOODBC needs *CHANGE authority to the CRLIM file. WOODBC is a member of three groups: DPTAR, DPTSM, and DPTMG. DPTAR is the first group profile (GRPPRF). DPTSM and DPTMG are supplemental group profiles (SUPGRPPRF). Figure 27 on page 181 shows the authorities for the CRLIM file:

```
                          Display Object Authority

Object . . . . . :   CRLIM          Owner  . . . . . :   OWNAR
  Library  . . . . . :   CUSTLIB       Primary group  . . . :   DPTAR
Object type    . . . :   *FILE         ASP device . . . . . :   *SYSBAS

 Object secured by authorization list  . . . . . . . . . . . . :   *NONE

                        Object
User        Group       Authority
OWNAR                   *ALL
DPTAR                   *CHANGE
DPTSM                   *USE
*PUBLIC                 *EXCLUDE
```

Figure 27. Authority for the CRLIM File

These are the authority checking steps:

1. Flowchart 1, step 1.
   a. Flowchart 2, step 1. Return to calling flowchart with insufficient authority.
2. Flowchart 1, step 2.
   a. Flowchart 3, steps 1 and 2. Object to check = CUSTLIB/CRLIM *FILE.
   b. Flowchart 3, step 3.
      1) Flowchart 4, step 1. WOODBC does not own the CRLIM file. Return to Flowchart 3 with no authority found.
   c. Flowchart 3, step 4.
      1) Flowchart 5, steps 1, 2 and 3. Public authority is not sufficient.
   d. Flowchart 3, step 5.
   e. **Flowchart 3, step 6**. WOODBC does not have any authority to the CRLIM file.
   f. Flowchart 3, steps 7 and 8. The CRLIM file is not secured by an authorization list. Return to Flowchart 1 with no authority found.
3. Flowchart 1, steps 3 and 4. The first group for WOODBC is DPTAR.
   a. Flowchart 6, steps 1 and 2. Object to check = CUSTLIB/CRLIM *FILE.
   b. Flowchart 6, step 3.
      1) Flowchart 4, step 1. DPTAR does not own the CRLIM file. Return to Flowchart 6 with no authority found.
   c. Flowchart 6, steps 4 and 5. Authorized. DPTAR is the primary group and has sufficient authority.

Case 11: Combining Authorization Methods

WAGNERB needs *ALL authority to the CRLIMWRK file. WAGNERB is a member of these groups: DPTSM, DPT702, and DPTAR. WAGNERB's first group (GRPPRF) is DPTSM. Figure 28 on page 182 shows the authority for the CRLIMWRK file.

```
                          Display Object Authority

Object . . . . . :   CRLIMWRK        Owner  . . . . . :   OWNAR
  Library  . . . . . :   CUSTLIB        Primary group  . . . :   *NONE
Object type    . . . :   *FILE          ASP device . . . . . :   *SYSBAS

Object secured by authorization list  . . . . . . . . . . . . :   CRLST1

                        Object
User        Group       Authority
OWNAR                   *ALL
DPTSM                   *USE
WILSONJ                 *EXCLUDE
*PUBLIC                 *USE
```

Figure 28. Authority for CRLIMWRK File

The CRLIMWRK file is secured by the CRLST1 authorization list. Figure 29 shows the authority for the CRLST1 authorization list.

```
                        Display Authorization List

Object . . . . . :   CRLST1        Owner  . . . . . :   OWNAR
  Library  . . . . . :     QSYS        Primary Group  . . . :   DPTAR

                        Object    List
User        Group       Authority  Mgt
OWNAR                   *ALL        X
DPTAR                   *ALL
*PUBLIC                 *EXCLUDE
```

Figure 29. Authority for the CRLST1 Authorization List

This example shows many of the possibilities for authority checking. It also demonstrates how using too many authority options for an object can result in poor performance.

Following are the steps required to check WAGNERB's authority to the CRLIMWRK file:

1. Flowchart 1, step 1.
   a. Flowchart 2, step 1.
2. Flowchart 1, step 2.
   a. Flowchart 3, steps 1 and 2. Object to check = CUSTLIB/CRLIMWRK *FILE.
   b. Flowchart 3, step 3.
      1) Flowchart 4, step 1. WAGNERB does not own the CRLIMWRK file. Return to Flowchart 3 with no authority found.
   c. Flowchart 3, step 4.
      1) Flowchart 5, steps 1 and 2. WILSONJ has *EXCLUDE authority, which is less than the public authority of *USE.

d. Flowchart 3, steps 5 and 6 (**first search of private authorities**). WAGNERB does not have private authority.

e. Flowchart 3, steps 7 and 9. Object to check = CRLST1 *AUTL.

f. Flowchart 3, step 3.

   1) Flowchart 4, step 1. WILSONJ does not own CRLST1. Return to Flowchart 3 with no authority found.

g. Flowchart 3, steps 4 and 5.

h. Flowchart 3, step 6 (**second search of private authorities**). WAGNERB does not have private authority to CRLST1.

i. Flowchart 3, steps 7 and 8. Object to check = CUSTLIB/CRLIMWRK *FILE.

3. Flowchart 1, steps 3 and 4. WAGNERB's first group profile is DPTSM.

   a. Flowchart 6, steps 1 and 2. Object to check = CUSTLIB/CRLIMWRK *FILE.

   b. Flowchart 6, step 3.

      1) Flowchart 4, step 1. DPTSM does not own the CRLIMWRK file. Return to Flowchart 6 with no authority found.

   c. Flowchart 6, step 4. DPTSM is not the primary group for the CRLIMWRK file.

   d. Flowchart 6, step 6 (**third search of private authorities**). DPTSM has *USE authority to the CRLIMWRK file, which is not sufficient.

   e. Flowchart 6, step 6 continued. *USE authority is added to any authorities already found for WAGNERB's groups (none). Sufficient authority has not yet been found.

   f. Flowchart 6, steps 9 and 10. WAGNERB's next group is DPT702.

   g. Flowchart 6, steps 1 and 2. Object to check = CUSTLIB/CRLIMWRK *FILE.

   h. Flowchart 6, step 3.

      1) Flowchart 4, step 1. DPT702 does not own the CRLIMWRK file. Return to Flowchart 6 with no authority found.

   i. Flowchart 6, step 4. DPT702 is not the primary group for the CRLIMWRK file.

   j. Flowchart 6, step 6 (**fourth search of private authorities**). DPT702 has no authority to the CRLIMWRK file.

   k. Flowchart 6, steps 7 and 8. Object to check = CRLST1 *AUTL

   l. Flowchart 6, step 3.

      1) Flowchart 5, step 1. DPT702 does not own the CRLST1 authorization list. Return to Flowchart 6 with no authority found.

   m. Flowchart 6, steps 4 and 6. (**fifth search of private authorities**). DPT702 has no authority to the CRLST1 authorization list.

   n. Flowchart 6, steps 7, 9, and 10. DPTAR is WAGNERB's next group profile.

   o. Flowchart 6, steps 1 and 2. Object to check = CUSTLIB/CRLIMWRK *FILE.

   p. Flowchart 6, step 3.

      1) Flowchart 4, step 1. DPTAR does not own the CRLIMWRK file. Return to Flowchart 6 with no authority found.

   q. Flowchart 6, steps 4 and 6. (**sixth search of private authorities**). DPTAR has no authority to the CRLIMWRK file.

   r. Flowchart 6, steps 7 and 8. Object to check = CRLST1 *AUTL

s. Flowchart 6, step 3.
   1) Flowchart 4, step 1. DPTAR does not own the CRLST1 authorization list. Return to Flowchart 6 with no authority found.

t. Flowchart 6, steps 4 and 5. Authorized. DPTAR is the primary group for the CRLST1 authorization list and has *ALL authority.

Result: WAGNERB is authorized to perform the requested operation using DPTAR's primary group authority to the CRLIST1 authorization list.

Analysis: This example demonstrates poor authority design, both from a management and performance standpoint. Too many options are used, making it difficult to understand, change, and audit. Private authorities are searched 6 separate times, which may cause noticeable performance problems:

Profile	Object	Type	Result
WAGNERB	CRLIMWRK	*FILE	No authority found
WAGNERB	CRLST1	*AUTL	No authority found
DPTSM	CRLIMWRK	*FILE	*USE authority (insufficient)
DPT702	CRLIMWRK	*FILE	No authority found
DPT702	CRLST1	*AUTL	No authority found
DPTAR	CRLIMWRK	*FILE	No authority found

Changing the sequence of WAGNERB's group profiles would change the performance characteristics of this example. Assume DPTAR is WAGNERB's first group profile (GRPPRF). The system would search private authorities 3 times before finding DPTAR's primary group authority to the CRLST1 authorization list.

- WAGNERB authority for CRLIMWRK file
- WAGNERB authority for CRLST1 authorization list
- DPTAR authority for CRLIMWRK file

Careful planning of group profiles and authorization lists is essential to good system performance.

Authority Cache

In Version 3, Release 7, the system creates an authority cache for a user the first time the user accesses an object. Each time the object is accessed, the system looks for authority in the user's cache before looking at the users's profile. This results in a faster check for private authority.

The authority cache contains up to 32 private authorities to objects and up to 32 private authorities to authorization lists. The cache is updated when a user authority is granted or revoked. All user caches are cleared when the system IPL is performed.

While limited use of private authorities is recommended, the cache offers flexibility. For example, you can choose how to secure objects with less concern about the impact on system performance. This is especially true if users access the same objects repeatedly.

Chapter 6. Work Management Security

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* book.

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 users 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. The user profile that is specified on the sign-on display must have *OBJOPR, and *CHANGE authority to itself.
2. The user's authority to use the workstation is checked. See "Workstations" on page 187 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 any of these objects does not exist or the user does not have adequate authority, 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 an object does not exist or 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. See 369 for the authority that is required to change parameters for a job.

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 30 on page 188 shows the logic for determining whether a user is allowed to sign on at a device:

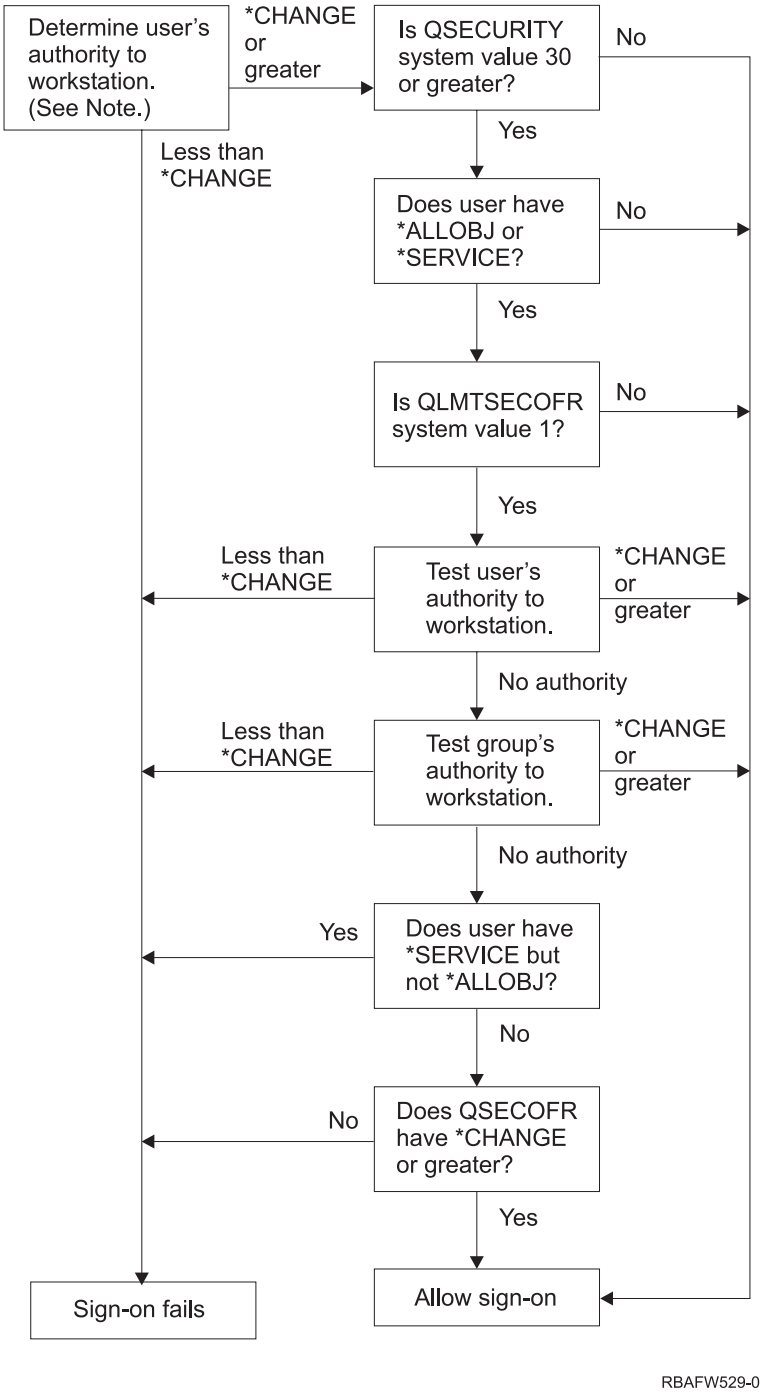


Figure 30. Authority Checking for Workstations

Note: Normal authority checking is performed to determine whether the user has at least *CHANGE authority to the device description. *CHANGE authority may be found by using the following:

- *ALLOBJ special authority from the user profile, group profile, or supplemental group profiles.
- Private authority to the device description in the user profile, the group profile, or supplemental group profiles.
- Authority to an authorization list used to secure the device description.
- Authority to an authorization list used to secure the public authority.

Authority checking for the device description is done before any programs are in the program stack for the job; therefore, adopted authority does not apply.

Description of Authority Checking for Workstations

The system determines the user's authority to the workstation. (See note 1) If the authority is less than *CHANGE then the sign-on fails. If the authority is *CHANGE or greater than the system check to see it the security level on the system is 30 or higher. If it is not, then the user is allowed to sign-on.

If the security level is 30 or higher, the system checks if the user has *ALLOBJ or *SERVICE special authorities. If the user does not have either of these special authorities, then sign-on is allowed.

If the user has either *ALLOBJ or *SERVICE special authorities, then the system checks if the QLMTSECOFR system value is set to 1. If it is not set to 1, then sign-on is allowed.

If the QLMTSECOFR system value is set to 1, then the system will test the user's authority to the workstation. If the user's authority is *CHANGE or higher, then sign-on is allowed. If the user's authority is less than *CHANGE, sign-on fails. If the user has no authority to the workstation, the system checks the user's group authority to the workstation.

If the user's group authority is *CHANGE or higher, then sign-on is allowed. If the user's group authority is less than *CHANGE, sign-on fails. If the user has no authority to the workstation, the system checks whether or not the user has *SERVICE but not *ALLOBJ special authority.

If the user has *SERVICE but not *ALLOBJ special authority, then sign-on fails. If the user does have *SERVICE but not *ALLOBJ special authority, then the system checks if QSECOFR has *CHANGE or higher.

If QSECOFR does not have*CHANGE or higher, then sign-on fails. If QSECOFR has *CHANGE or higher, then sign-on is allowed.

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 the QSRV or QSRVBAS profile 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.

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, most devices are owned by the QPGMR profile. Devices created by the QLUS program (*APPC type devices) are owned by the QSYS 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.

Signon screen display file

The system administrator can change the system signon display to add text or company logo to the display. Care must be taken to make sure the field names or buffer lengths of the display file are not changed when adding text to the display file. Changing the field names or buffer lengths may cause signon to fail.

Changing the signon screen display

The source code for the signon display file is shipped with the operating system. The source is shipped in file QSYS/QAWTSSRC. This source code can be changed to add text to the signon screen display. Field names and buffer lengths should not be changed.

Display file source for the Signon screen

The source for the signon display file is shipped as a member (QDSIGNON or QDSIGNON2) in the QSYS/QAWTSSRC physical file. QDSIGNON contains the source for the signon screen source used when system value QPWDLVL is set to 0 or 1. Member QDSIGNON2 contains the signon screen source used when the system value QPWDLVL is set to 2 or 3.

The file QSYS/QAWTSSRC is **deleted and restored** each time the OS/400 operating system is installed. If you plan to create your own version of the signon screen, then you should first copy the appropriate source file member, either QDSIGNON or QDSIGNON2, to your own source file and make changes to the copy in your source file.

Changing the signon display file

To change the format of the Signon display:

1. Create a changed signon display file.

   A hidden field in the display file named UBUFFER can be changed to manage smaller fields. UBUFFER is 128 bytes long and is stated as the last field in the display file. This field can be changed to function as an input/output buffer so the data specified in this field of the display will be available to application

programs when the interactive job is started. You can change the UBUFFER field to contain as many smaller fields as you need if the following requirements are met:

- The new fields must follow all other fields in the display file. The location of the fields on the display does not matter as long as the order in which they are put in the data description specifications (DDS) meets this requirement.
- The length must total 128. If the length of the fields is more than 128, some of the data will not be passed.
- All fields must be input/output fields (type B in DDS source) or hidden fields (type H in DDS source).

2. The order in which the fields in the signon display file are declared must not be changed. The position in which they are shown on the display can be changed. Do not change the existing field names in the source for the signon screen display file.
3. Do not change the total size of the input or output buffers. Serious problems can occur if the order or size of the buffers are changed.
4. Do not use the data descriptions specifications (DDS) help function in the signon display file.
5. Change a subsystem description to use the changed display file instead of the system default of QSYS/QDSIGNON. You can change the subsystem descriptions for subsystems that you want to use the new display. To change the subsystem description:
   a. Use the Change Subsystem Description (CHGSBSD) command.
   b. Specify the new display file on the SGNDSPF parameter.
   c. Use a test version of a subsystem to verify that the display is valid before attempting to change the controlling subsystem.
6. Test the change.
7. Change the other subsystem descriptions.

Notes:

1. The buffer length for the display file must be 318. If it is less than 318, the subsystem uses the default sign-on display, QDSIGNON in library QSYS when system value QPWDLVL is 0 or 1 and QDSIGNON2 in library QSYS when QPWDLVL is 2 or 3.
2. The copyright line cannot be deleted.

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 levels 30 and higher, the system logs an entry (type AF, sub-type S) in the audit journal, if default sign-on is attempted and the auditing function is active. 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. See "Signing On without a User ID and Password" on page 16 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. "Network Attributes" on page 200 discusses the network attributes that are important for security.

Job Descriptions

A job description is a valuable tool for security and work management. You can also 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 191 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.

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 *EXCLUDE.

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, and has *USE authority to the QBATCH job description, 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 also required.

System Operator Message Queue

The iSeries Operational Assistant (ASSIST) menu provides an option to manage your system, users, and devices. The Manage Your System, Users, and Devices menu provides an option 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 392.) 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.

Attention: All jobs need the ability to add new messages to the QSYSOPR message queue. Do not make the public authority to QSYSOPR *EXCLUDE.

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 108 on page 194 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 108. 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 250 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.

"Library Security and Library Lists" on page 123 provides some general information about the issues associated with library lists. This topic gives more specific examples of the possible exposures and how to avoid them.

Following are two examples of how changes to a library list might break security requirements:

Change in Function

Figure 31 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.

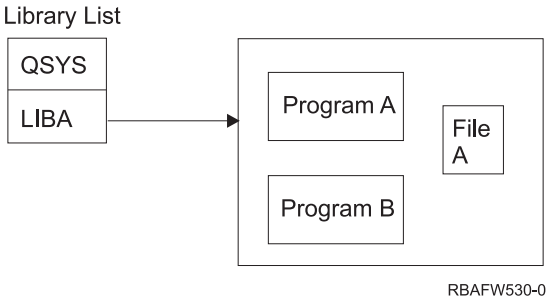


Figure 31. Library List–Expected Environment

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:

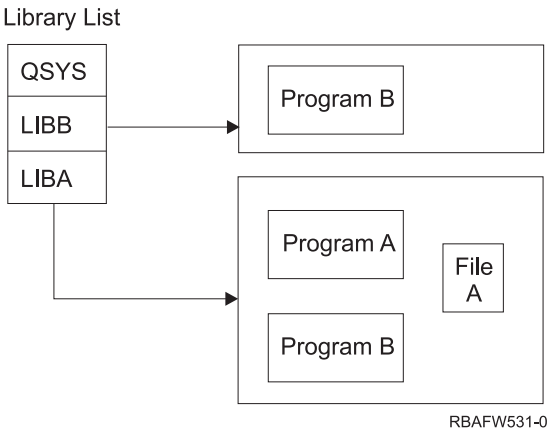


Figure 32. Library List–Actual Environment

Unauthorized Access to Information

Assume Program A in Figure 31 on page 194 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 are shipped with public authority *ADD for production reasons. Regularly monitor what objects (particularly programs, source files, 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 216.

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 CMDX 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), Create Menu (CRTMNU), Change Menu (CHGMNU), Create Command (CRTCMD), and Change Command (CHGCMD) commands.
- Use the technique described in "Controlling the User Library List" on page 215 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 217 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 215 for an example of this technique.

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 *Printer Device Programming* book 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 Queue Description (WRKOUTQD) command.

Attention: A user with *SPLCTL special authority can perform all functions on all entries, regardless of how the output queue is defined. Some parameters on the output queue allow a user with *JOBCTL special authority to view the contents of entries on the output queue.

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

***NO**	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.
***OWNER**	Only the owner of a spooled file or a user with *SPLCTL (spool control) 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

***OWNER**	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

***YES**	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.

Output Queue and Parameter Authorities Required for Printing

Table 109 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. For more information see "Writer Commands" on page 440.

The authority and output queue parameters for all commands associated with spooled files are listed on "Spooled File Commands" on page 427. Output queue commands are listed on "Output Queue Commands" on page 404.

Attention: 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 109. Authority Required to Perform Printing Functions

	Output Queue Parameters			Output Queue	Special
Printing Function	**DSPDTA**	**AUTCHK**	**OPRCTL**	**Authority**	**Authority**
Add spooled files to queue [1]				*READ	None
			*YES		*JOBCTL
View list of spooled files				*READ	None
(WRKOUTQ command [2])			*YES		*JOBCTL
Display, copy, or send spooled files	*YES			*READ	None
(DSPSPLF, CPYSPLF, SNDNETSPLF, SNDTCPSP[2])	*NO	*DTAAUT		*READ, *ADD, *DLT	None
	*NO	*OWNER		Owner [3]	None
	*YES		*YES		*JOBCTL
	*NO		*YES		*JOBCTL
	*OWNER				
Change, delete, hold, and release spooled file		*DTAAUT		*READ, *ADD, *DLT	None
(CHGSPLFA, DLTSPLF, HLDSPLF, RLSSPLF [2])					
		*OWNER		Owner [3]	None
			*YES		*JOBCTL
Change, clear, hold, and release output queue		*DTAAUT		*READ, *ADD, *DLT	None
(CHGOUTQ, CLROUTQ, HLDOUTQ, RLSOUTQ [2])					
		*OWNER		Owner [3]	None
			*YES		*JOBCTL
Start a writer for the queue		*DTAAUT		*CHANGE	None
(STRPRTWTR, STRRMTWTR [2])			*YES		*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] *CHGOUTQ requires *OBJMGT authority to the output queue, in addition to *READ, *ADD, and *DLT authorities.

Examples: Output Queue

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) AUTCHK(*OWNER) 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) AUTCHK(*OWNER) 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)
```

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 topics that follow:

Job action (JOBACN)

Client request access (PCSACC)

DDM request access (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 *SNA Distribution Services* book.

Client Request Access (PCSACC) Network Attribute

The PCSACC network attribute determines how the iSeries Access for Windows 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 iSeries system, not whether the personal computer can use workstation emulation.

Note: PCSACC network attribute controls only the DOS and OS/2® clients. This attribute has no effect on any other Client Access clients.

Possible Values for PCSACC:

***REJECT**	iSeries Access rejects every request from the personal computer to access objects on the iSeries system. An error message is sent to the PC application.
***OBJAUT**	The iSeries Access 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.
***REGFAC**	The system uses the system's registration facility to determine which exit program (if any) to run. If no exit program is defined for an exit point and this value is specified, *OBJAUT is used.
qualified- program- name	The iSeries Access 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 iSeries Access 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 iSeries Access 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 iSeries Access 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 iSeries 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.
- Restricting authority to commands used by iSeries Access.
- 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 iSeries Access licensed program. Using an exit program to verify all requests is the only adequate protection measure.

The iSeries Access 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

For additional information on Client Access, refer to the Information Center (see "Prerequisite and related information" on page xvi for details).

DDM Request 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 or DRDA® 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 Information Center (see "Prerequisite and related information" on page xvi for details).

Save and Restore Operations

The ability to save objects from your system or restore objects to your system represents an exposure to your organization.

For example, programmers often have *OBJEXIST authority to programs because this authority is required to recompile a program (and delete the old copy). *OBJEXIST authority is also required to save an object. Therefore, the typical programmer can make a tape copy of your programs, which may represent a substantial financial investment.

A user with *OBJEXIST authority to an object can also restore a new copy of an object over an existing object. In the case of a program, the restored program might have been created on a different system. It might perform different functions. For example, assume the original program worked with confidential data. The new version might perform the same functions, but it might also write a copy of confidential information to a secret file in the programmer's own library. The programmer does not need authority to the confidential data because the regular users of the program will be accessing the data.

Restricting Save and Restore Operations

You can control the ability to save and restore objects in several ways:

- Restrict physical access to save and restore devices, such as tape units, optical units, and diskette units.
- Restrict authority to the device descriptions objects for the save and restore devices. To save an object to a tape unit, you must have *USE authority to the device description for the tape unit.
- Restrict the save and restore commands. This allows you to control what is saved from your system and restored to your system through all interfaces - including save files. See "Example: Restricting Save and Restore Commands" for an example of how to do this. The system sets the restore commands to PUBLIC(*EXCLUDE) when you install your system.
- Only give *SAVSYS special authority to trusted users.

Example: Restricting Save and Restore Commands

Following is an example of the steps that you can use to restrict the save and restore commands on your system:

1. To create an authorization list that you can use to give authority to the commands to system operators, type the following:

   ```
   CRTAUTL AUTL(SRLIST) TEXT('Save and Restore List')
   AUT(*EXCLUDE)
   ```

2. To use the authorization list to secure the save commands, type the following:

   ```
   GRTOBJAUT OBJ(SAV*) OBJTYPE(*CMD) AUTL(SRLIST)
   ```

3. To ensure *PUBLIC authority comes from the authorization list, type the following:

   ```
   GRTOBJAUT OBJ(SAV*) OBJTYPE(*CMD) USER(*PUBLIC)
      AUT(*AUTL)
   ```

4. To use the authorization list to secure the restore commands, type the following:

   ```
   GRTOBJAUT OBJ(RST*) OBJTYPE(*CMD) AUTL(SRLIST)
   ```

5. To ensure *PUBLIC authority comes from the authorization list, type the following:

```
GRTOBJAUT OBJ(RST*) OBJTYPE(*CMD) USER(*PUBLIC)
   AUT(*AUTL)
```

6. Although system operators who are responsible for saving the system have *SAVSYS special authority, they must now be given explicit authority to the SAVxxx commands. You do this by adding the system operators to the authorization list:

   ```
   ADDAUTLE AUTL(SRLIST) USER(USERA USERB) AUT(*USE)
   ```

   Note: You may want your system operators to have authority only to the save commands. In that case, secure the save commands and the restore commands with two separate authorization lists.

7. To restrict the save and restore APIs and secure it with the authorization list, type the following commands:

   ```
   GRTOBJAUT OBJ(QSRSAVO) OBJTYPE(*PGM) AUTL(SRLIST)
   GRTOBJAUT OBJ(QSRSAVO) OBJTYPE(*PGM) USER(*PUBLIC)
      AUT(*AUTL)
   GRTOBJAUT OBJ(QSRLIB01) OBJTYPE(*SRVPGM) AUTL(SRLIST)
   GRTOBJAUT OBJ(QSRLIB01) OBJTYPE(*SRVPGM) USER(*PUBLIC)
      AUT(*AUTL)
   ```

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 System Security and Planning topic in the Information Center 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 Information Center (see "Prerequisite and related information" on page xvi for details). They can help you view your application from the perspective of a security administrator and understand what information you need to provide.

The Basic System Security and Planning topic in the Information Center also 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 33 on page 208 shows the relationships between user groups, applications, and libraries for the JKL Toy Company:

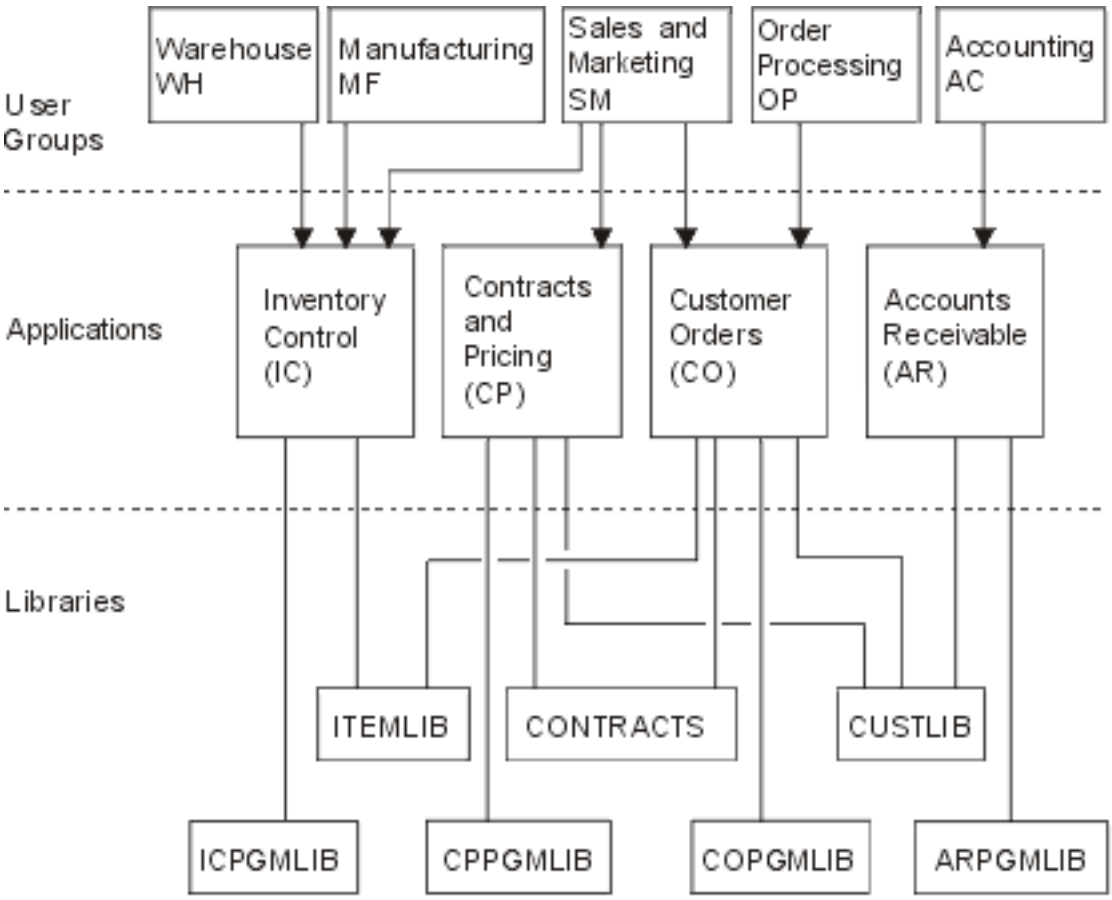


Figure 33. Example Applications

Description of graphic

This graphic shows how how five sets of user groups access applications and libraries on the system at JKL Toy Company. The user groups include Warehouse, Manufacturing, Sales and Marketing, Order Processing, and Accounting. The Warehouse, Manufacturing and Sales and Marketing user groups all can access the Inventory Control applications. The Sales and Marketing user group also has access to the Contracts and Pricing application and the Customer Order application. The Order Processing user group also can access the Customer Order application. The Accounting user group uses the Accounts Receivable application.

Overall Recommendations

The recommendations in this chapter and in the Basic System Security and Planning topic in the Information Center 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:

- Use resource security along with the methods available, such as limited capabilities in the user profile and restricting users to a set of menus, to protect information.

Attention: It is not sufficient to use only limited capabilities in the user profile and menu access control to secure your system if you use a product such as Client Access/400 or have communication lines attached to your system. You must use resource security to secure those objects you do not want accessible through these interfaces.

- 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.

Planning Password Level Changes

Changing password levels should be planned carefully. Operations with other systems may fail or users may not be able to sign on to the system if you haven't planned for the password level change adequately. Prior to changing the QPWDLVL system value, make sure you have saved your security data using the SAVSECDTA or SAVSYS command. If you have a current backup, you will be able to reset the passwords for all users' profiles if you need to return to a lower password level.

Products that you use on the system, and on clients with which the system interfaces, may have problems when the password level (QPWDLVL) system value is set to 2 or 3. Any product or client that sends passwords to the system in an encrypted form, rather than in the clear text a user enters on a sign-on screen, must be upgraded to work with the new password encryption rules for QPWDLVL 2 or 3. Sending the encrypted password is known as password substitution. Password substitution is used to prevent a password from being captured during transmission over a network. Password substitutes generated by older clients that do not support the new algorithm for QPWDLVL 2 or 3, even if the specific characters typed in are correct, will not be accepted. This also applies to any iSeries to iSeries peer access which utilizes the encrypted values to authenticate from one system to another.

The problem is compounded by the fact that some affected products (i.e. Java Toolbox) are provided as middleware. A third party product that incorporates a prior version of one of these products will not work correctly until rebuilt using an updated version of the middleware.

Given this and other scenarios, it is easy to see why careful planning is necessary before changing the QPWDLVL system value.

Considerations for changing QPWDLVL from 0 to 1

Password level 1 allows a system, which does not have a need to communicate with the Windows 95/98/ME iSeries Client Support for Windows Network Neighborhood (NetServer) product, to have the NetServer passwords eliminated from the system. Eliminating unnecessary encrypted passwords from the system increases the overall security of the system.

At QPWDLVL 1, all current, pre-V5R1 password substitution and password authentication mechanisms will continue to work. There is very little potential for breakage except for functions/services that require the NetServer password.

The functions/services that require the NetServer password include:

- iSeries Support for Windows Network Neighborhood, Windows 95/98/ME edition, (NetServer)

Considerations for changing QPWDLVL from 0 or 1 to 2

Password level 2 introduces the use of case sensitive passwords up to 128 characters in length (also called passphrases) and provides the maximum ability to revert back to QPWDLVL 0 or 1.

Regardless of the password level of the system, password level 2 and 3 passwords are created whenever a password is changed or a user signs on to the system. Having a level 2 and 3 password created while the system is still at password level 0 or 1 helps prepare for the change to password level 2 or 3.

Prior to changing QPWDLVL to 2, the system administrator should use the DSPAUTUSR or PRTUSRPRF TYPE(*PWDINFO) commands to locate all user profiles which do not have a password that is usable at password level 2. Depending on the profiles located, the administrator may wish to use one of the following mechanisms to have a password level 2 and 3 password added to the profiles.

- Change the password for the user profile using the CHGUSRPRF or CHGPWD CL command or the QSYCHGPW API. This will cause the system to change the password that is usable at password levels 0 and 1; and the system also creates two equivalent case sensitive passwords that are usable at password levels 2 and 3. An all uppercase and all lowercase version of the password is created for use at password level 2 or 3.

  For example, changing the password to C4D2RB4Y results in the system generating C4D2RB4Y and c4d2rb4y password level 2 passwords.
- Sign on to the system through a mechanism that presents the password in clear text (does not use password substitution). If the password is valid and the user profile does not have a password that is usable at password levels 2 and 3, the system creates two equivalent case sensitive passwords that are usable at password levels 2 and 3. An all uppercase and all lowercase version of the password is created for use at password level 2 or 3.

The absence of a password that is usable at password level 2 or 3 can be a problem whenever the user profile also does not have a password that is usable at password levels 0 and 1 or when the user tries to sign on through a product that uses password substitution. In these cases, the user will not be able to sign on when the password level is changed to 2.

If a user profile does not have a password that is usable at password levels 2 and 3, the user profile does have a password that is usable at password levels 0 and 1,

and the user signs on through a product that sends clear text passwords, then the system validates the user against the password level 0 password and creates two password level 2 passwords (as described above) for the user profile. Subsequent sign ons will be validated against the password level 2 passwords.

Any client/service which uses password substitution will not work correctly at QPWDLVL 2 if the client/service hasn't been updated to use the new password (passphrase) substitution scheme. The administrator should check whether a client/service which hasn't been updated to the new password substitution scheme is required.

The clients/services that use password substitution include:

- TELNET
- iSeries Access
- iSeries Host Servers
- QFileSrv.400
- iSeries NetServer Print support
- DDM
- DRDA
- SNA LU6.2

It is highly recommended that the security data be saved prior to changing to QPWDLVL 2. This can help make the transition back to QPWDLVL 0 or 1 easier if that becomes necessary.

It is recommended that the other password system values, such as QPWDMINLEN and QPWDMAXLEN not be changed until after some testing at QPWDLVL 2 has occurred. This will make it easier to transition back to QPWDLVL 1 or 0 if necessary. However, the QPWDVLDPGM system value must specify either *REGFAC or *NONE before the system will allow QPWDLVL to be changed to 2. Therefore, if you use a password validation program, you may wish to write a new one that can be registered for the QIBM_QSY_VLD_PASSWRD exit point by using the ADDEXITPGM command.

NetServer passwords are still supported at QPWDLVL 2, so any function/service that requires a NetServer password should still function correctly.

Once the administrator is comfortable with running the system at QPWDLVL 2, they can begin to change the password system values to exploit longer passwords. However, the administrator needs to be aware that longer passwords will have these effects:

- If passwords greater than 10 characters are specified, the password level 0 and 1 password is cleared. This user profile would not be able to signon if the system is returned to password level 0 or 1.
- If passwords contain special characters or do not follow the composition rules for simple object names (excluding case sensitivity), the password level 0 and 1 password is cleared.
- If passwords greater than 14 characters are specified, the NetServer password for the user profile is cleared.
- The password system values only apply to the new password level 2 value and do not apply to the system generated password level 0 and 1 password or NetServer password values (if generated).

Considerations for changing QPWDLVL from 2 to 3

After running the system at QPWDLVL 2 for some period of time, the administrator can consider moving to QPWDLVL 3 to maximize his password security protection.

At QPWDLVL 3, all NetServer passwords are cleared so a system should not be moved to QPWDLVL 3 until there is no need to use NetServer passwords.

At QPWDLVL 3, all password level 0 and 1 passwords are cleared. The administrator can use the DSPAUTUSR or PRTUSRPRF commands to locate user profiles which don't have password level 2 or 3 passwords associated with them.

Changing to a lower password level

Returning to a lower QPWDLVL value, while possible, is not expected to be a completely painless operation. In general, the mind set should be that this is a one-way trip from lower QPWDLVL values to higher QPWDLVL values. However, there may be cases where a lower QPWDLVL value must be reinstated.

The following sections each discuss the work required to move back to a lower password level.

Considerations for changing from QPWDLVL 3 to 2

This change is relatively easy. Once the QPWDLVL is set to 2, the administrator needs to determine if any user profile is required to contain NetServer passwords or password level 0 or 1 passwords and, if so, change the password of the user profile to an allowable value.

Additionally, the password system values may have to be changed back to values compatible with NetServer and password level 0 or 1 passwords, if those passwords are needed.

Considerations for changing from QPWDLVL 3 to 1 or 0

Because of the very high potential for causing problems for the system (like no one can sign on because all of the password level 0 and 1 passwords have been cleared), this change is not supported directly. To change from QPWDLVL 3 to QPWDLVL 1 or 0, the system must first make the intermediary change to QPWDLVL 2.

Considerations for changing from QPWDLVL 2 to 1

Prior to changing QPWDLVL to 1, the administrator should use the DSPAUTUSR or PRTUSRPRF TYPE(*PWDINFO) commands to locate any user profiles that do not have a password level 0 or 1 password. If the user profile will require a password after the QPWDLVL is changed, the administrator should ensure that a password level 0 and 1 password is created for the profile using one of the following mechanisms:

- Change the password for the user profile using the CHGUSRPRF or CHGPWD CL command or the QSYCHGPW API. This will cause the system to change the password that is usable at password levels 2 and 3; and the system also creates an equivalent uppercase password that is usable at password levels 0 and 1. The system is only able to create the password level 0 and 1 password if the following conditions are met:
  - The password is 10 characters or less in length.
  - The password can be converted to uppercase EBCDIC characters A-Z, 0-9, @, #, $, and underscore.
  - The password does not begin with a numeric or underscore character.

For example, changing the password to a value of RainyDay would result in the system generating a password level 0 and 1 password of RAINYDAY. But changing the the password value to Rainy Days In April would cause the system to clear the password level 0 and 1 password (because the password is too long and it contains blanks).

No message or indication is produced if the password level 0 or 1 password could not be created.

- Sign on to the system through a mechanism that presents the password in clear text (does not use password substitution). If the password is valid and the user profile does not have a password that is usable at password levels 0 and 1, the system creates an equivalent uppercase password that is usable at password levels 0 and 1. The system is only able to create the password level 0 and 1 password if the conditions listed above are met.

The administrator can then change QPWDLVL to 1. All NetServer passwords are cleared when the change to QPWDLVL 1 takes effect (next IPL).

Considerations for changing from QPWDLVL 2 to 0

The considerations are the same as for changing from QPWDLVL 2 to 1 except that all NetServer passwords are retained when the change takes effect.

Considerations for changing from QPWDLVL 1 to 0

After changing QPWDLVL to 0, the administrator should use the DSPAUTUSR or PRTUSRPRF commands to locate any user profiles that do not have a NetServer password. If the user profile requires a NetServer password, it can be created by changing the user's password or signing on through a mechanism that presents the password in clear text.

The administrator can then change QPWDLVL to 0.

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 123 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 33 on page 208 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. That is, application libraries should have public authority of *USE or *EXCLUDE unless users need to create objects directly into the library.
- Library lists are controlled.

Planning Applications to Prevent Large Profiles

Because of the potential impacts to performance and security, IBM **strongly recommends** the following to avoid profiles from becoming too full:

- Do not have one profile own everything on your system.

  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. Also, information about private authorities is spread among several profiles, which improves performance. By using several owner profiles, you can prevent a profile from becoming too large because of too many objects. 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. Applications owned by IBM-supplied user profiles can also impact performance for commands, such as, CHKOBJITG and WRKOBJOWN.
- Use authorization lists to secure objects.

  If you are granting private authorities to many objects for several users, you should consider using an authorization list to secure the objects. Authorization lists will cause one private authority entry for the authorization list in the user's profile rather than one private authority entry for each object. In the object owner's profile, authorization lists cause an authorized object entry for every

user granted authority to the authorization list rather than an authorized object entry for every object multiplied by the number of users that are granted the private authority.

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 "Library Lists" on page 193 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(2750)
          DCL      &CURLIB  *CHAR LEN(10)
          DCL      &ERROR *LGL
          DCL      &CMD *CHAR LEN(2800)
          MONMSG   MSGID(CPF0000)  +
                   EXEC(GOTO SETERROR)
          RTVJOBA  USRLIBL(&USRLIBL) +
                   CURLIB(&CURLIB)
          IF COND(&CURLIB=('*NONE')) +
             THEN(CHGVAR &CURLIB '*CRTDFT  ')
          CHGLIBL  LIBL(QGPL) 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 2800)
                   IF       &ERROR SNDPGMMSG MSGID(CPF9898) +
                            MSGF(QCPFMSG) MSGTYPE(*ESCAPE) +
                            MSGDTA('The xxxx error occurred')
                   ENDPGM
```

Figure 34. 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 34 on page 215, 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 35 on page 217 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 35. Format for Describing Library Security

Planning Menus

Menus are a good method for providing controlled access 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 an access control 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 from menus. The topic “Using Adopted Authority in Menu Design” on page 218 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 222 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 36. 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 135 describes how adopted authority works. "Flowchart 8: How Adopted Authority Is Checked" on page 169 describes how the system checks for adopted authority.

Figure 37 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 37. 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 to 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 38. 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 39 shows the QRYSTART program:

```
PGM
ADDLIBLE ITEMLIB
ADDLIBLE CUSTLIB
STRQRY
RMVLIBLE ITEMLIB
RMVLIBLE CUSTLIB
ENDPGM
```

Figure 39. Sample Program for Query with Adopted Authority

The menu system uses three types of user profiles, shown in Table 110. Table 111 describes the objects used by the menu system.

Table 110. 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 111. 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		

Table 111. Objects Used by Menu System (continued)

Object Name	Owner	Public Authority	Private Authorities	Additional Information
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 8: Adopted Authority without Private Authority" on page 178 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 139 provides more information about ignoring adopted authority. "Flowchart 8: How Adopted Authority Is Checked" on page 169 describes how the system checks for adopted authority.

Figure 40 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 40. Sample Application Menu with Query

The authority information for the QRYSTART program is the same as shown in Table 111 on page 219. 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 37 on page 218) 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 36 on page 218) 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 41 shows a sample format for providing this information.

Menu name: MENU1 Library: QGPLOption 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 41. 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.

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 112 shows the commands associated with the menu options:

Table 112. 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	Start System Request at Previous System (TFRPASTHR). (See note below.)
11	Transfer to previous system (TFRPASTHR). (See note below.)
12	Display 3270 emulation options (See note below.)
13	Start System Request at Home System (TFRPASTHR). (See note below.)
14	Transfer to Home System (TFRPASTHR). (See note below.)
15	Transfer to End System (TFRPASTHR). (See note below.)
50	End Request on Remote System (ENDRDBRQS). (See note below.)
80	Disconnect Job (DSCJOB)
90	Sign-Off (SIGNOFF)

Notes:

1. Options 10, 11, 13, 14, and 15 are displayed only if display station pass-through has been started with the Start Pass-Through (STRPASTHR) command. Option 10, 13, and 14 are only displayed on the target system.
2. Option 12 is only displayed when 3270 emulation is active.
3. Option 50 is displayed only if a remote jobs is active.
4. Some of the options have restrictions for the System/36 environment.

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 218 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.

Securing Logical Files

Resource security on the system supports field-level security of a file. You can also use logical files to protect specific fields or records in a file. See the DB2 Universal Database® for iSeries topic in the Information Center for more information. See "Prerequisite and related information" on page xvi for details.

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. Providing access to the information defined by a logical file requires data authority to both the logical file and the associated physical files.

Figure 42 shows an example of a physical file and three different logical files associated with it.

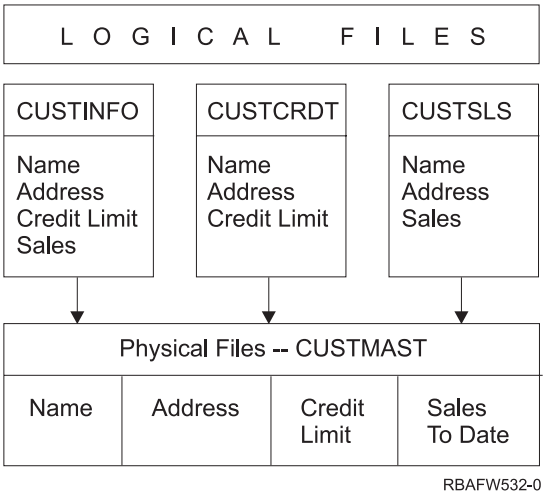


Figure 42. Using a Logical File for Security

Members of the sales department (group profile DPTSM) are allowed to view all fields, but they cannot change the credit limit. Members of the accounts receivable department (group profile DPTAR) are allowed to view all fields, but they cannot change the sales field. The authority to the physical file looks like this:

Table 113. Physical File Example: CUSTMAST File

	Users
Authority	***PUBLIC**
Object Authorities	
*OBJOPR	
*OBJMGT	
*OBJEXIST	
*OBJALTER	
*OBJREF	
Data Authorities	
*READ	X
*ADD	X
*UPD	X
*DLT	X
*EXECUTE	X
*EXCLUDE	

The public should have all data rights but no operational rights to the CUSTMAST physical file. The public cannot access the CUSTMAST file directly because *OBJOPR authority is required to open a file. The public's authority makes all the data rights potentially available to users of the logical file.

Authority to the logical files looks like this:

```
                          Display Object Authority

Object . . . . . :   CUSTINFO      Owner  . . . . . :   OWNAR
  Library  . . . . :   CUSTLIB       Primary group  . . :   *NONE
Object type  . . . . :   *FILE        ASP device . . . . . :   *SYSBAS

Object secured by authorization list  . . . . . :   *NONE

                        Object
User        Group       Authority
*PUBLIC                 *USE
```

```
                          Display Object Authority

Object . . . . . :   CUSTCRDT      Owner  . . . . . :   OWNAR
  Library  . . . . :   CUSTLIB       Primary group  . . :   DPTAR
Object type  . . . . :   *FILE        ASP device . . . . . :   *SYSBAS

Object secured by authorization list  . . . . . :   *NONE

                        Object
User        Group       Authority
DPTAR                   *CHANGE
*PUBLIC                 *USE
```

```
                          Display Object Authority

Object . . . . . :   CUSTSLS       Owner  . . . . . :   OWNSM
  Library  . . . . :   CUSTLIB       Primary group  . . :   DPTSM
Object type  . . . . :   *FILE        ASP device . . . . . :   *SYSBAS

Object secured by authorization list  . . . . . :   *NONE

                        Object
User        Group       Authority
DPTSM                   *CHANGE
*PUBLIC                 *USE
```

Making the group profile, such as DPTSM, the primary group for the logical file is not necessary for this authority scheme to work. However, using primary group authority eliminates searching private authorities for both the user attempting to access the file and the user's group. "Case 2: Using Primary Group Authority" on page 174 shows how using primary group authority affects the authority checking process.

You can specify data authorities for logical files beginning with V3R1 of the OS/400 licensed program. When you move to V3R1 from an earlier version, the system converts your logical files when the system is installed. The first time a logical file is accessed, the system gives it all data authorities.

To use logical files as a security tool, do this:

- Grant all data authorities to the underlying physical files.
- Revoke *OBJOPR from the physical files. This prevents users from accessing the physical files directly.
- Grant the appropriate data authorities to logical files. Revoke any authorities you do not want.
- Grant *OBJOPR to the logical files.

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.

File Security and SQL

Structured Query Language (SQL) uses cross-reference files to keep track of database files and their relationships. These files are collectively referred to as the SQL catalog. Public authority to the SQL catalog is *READ. This means that any user who has access to the SQL interface can display the names and text descriptions for all files on your system. The SQL catalog does not affect the normal authority required to access the contents of database files.

Care should be taken when using a CL program that adopts authority to start SQL or Query Manager. Both of these query programs allow users to specify a file name. The user can, therefore, access any file that the adopted profile has authority to.

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 has a private authority to one object, the authorization list. This gives the user authority to all the objects on the list. Reducing the number of private authorities in the system has the following advantages:
  - Reduces the size of user profiles.
  - Improves the performance when saving the system (SAVSYS) or saving the security data (SAVSECDTA).

- Authorization lists provide a good way to secure files. If you use private authorities, each user will have a private authority for each file member. If you use an authorization list, each user will have only one authority. Also, files that are open cannot have authority granted to the file or revoked from the file. If you secure the file with an authorization list, you can change the authorities, even when the file is open.
- 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.

Advantages of Using an Authorization List

From a security management view, an authorization list is the preferred method to manage objects that have the same security requirements. Even when there are only a few objects that would be secured by the list, there is still an advantage to using an authorization list instead of using private authorities on the object. Because the authorities are in one place (the authorization list), it is easier to change who is authorized to the objects. It is also easier to secure any new objects with the same authorities as the existing objects.

If you use authorization lists, then you should not have private authorities on the object. Two searches of the user's private authorities are required during the authority checking if the object has private authorities and the object is also secured by an authorization list. The first search is for the private authorities on the object; the second search is for the private authorities on the authorization list. Two searches require use of system resources; therefore, the performance can be impacted. If you use only the authorization list, only one search is performed. Also, because of the use of authority caching with the authorization list, the performance for the authority check will be the same as it is for checking only private authorities on the object.

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)
   ```

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 one of the following occurs:

- Another profile designates it as a group profile
- You assign a group identification number (gid) to it.

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 and assigns a gid to it.

Planning Primary Groups for Objects

Any object on the system can have a primary group. Primary group authority can provide a performance advantage if the primary group is the first group for most users of an object.

Often, one group of users is responsible for some information on the system, such as customer information. That group needs more authority to the information than other system users. By using primary group authority, you can set up this type of authority scheme without affecting the performance of authority checking. “Case 2: Using Primary Group Authority” on page 174 shows an example of this.

Planning Multiple Group Profiles

A user can be a member of up to 16 groups: the first group (GRPPRF parameter in the user profile) and 15 supplemental groups (SUPGRPPRF parameter in the user profile). By using group profiles, you can manage authority more efficiently and reduce the number of individual private authorities for objects. However, the misuse of group profiles can have a negative impact on the performance of authority checking.

Follow these suggestions when using multiple group profiles:

- Try to use multiple groups in combination with primary group authority and eliminate private authority to objects.
- Carefully plan the sequence in which group profiles are assigned to a user. The user’s first group should relate to the user’s primary assignment and the objects used most often. For example, assume a user called WAGNERB does inventory work regularly and does order entry work occasionally. The profile needed for inventory authority (DPTIC) should be WAGNERB’s first group. The profile needed for order entry work (DPTOE) should be WAGNERB’s first supplemental group.

Note: The sequence in which private authorities are specified for an object has no effect on authority checking performance.

- If you plan to use multiple groups, study the authority checking process described in "How the System Checks Authority" on page 156. Be sure you understand how using multiple groups in combination with other authority techniques, such as authorization lists, may affect your system performance.

Accumulating Special Authorities for Group Profile Members

Special authorities of group profiles are available to the members of that group. User profiles that are members of one or more groups have their own special authorities, plus the special authorities of any group profiles for which the user is a member. Special authorities are cumulative for users who are members of multiple groups. For example, assume that profile GROUP1 has *JOBCTL, profile GROUP3 has *AUDIT, and profile GROUP16 has *IOSYSCFG special authorities. A user profile that has all three profiles as its group profiles has *JOBCTL, *AUDIT, and *IOSYSCFG special authorities.

Note: ATTENTION

If a group member owns a program, the program adopts only the authority of the owner. The authorities of the group are **not** adopted.

Using an Individual Profile 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 that have similar security requirements. Authorization lists are used to secure objects with similar security requirements. Table 114 on page 231 shows the characteristics of the two methods:

Table 114. 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	Yes
Private authority overrides other authority	Yes	Yes
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 [2]

[1] The group profile can be given authority when an object is created by using the GRPAUT parameter in the profile of the user creating an object.

[2] Primary group authority is saved with the 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 **all** special authorities to programmers. However, if you must give programmers special authorities, give them **only** the special authority required to perform the jobs or tasks assigned 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 129 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.

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.

Planning the Use of Validation List Objects

Validation list objects are a new object type in Version 4, Release 1 that provide a method for applications to securely store user authentication information.

For example, the Internet Connection Server (ICS) uses validation lists to implement the concept of an **Internet user**. For Version 4, Release 1, the ICS can perform **basic authentication** before a web page is served. Basic authentication requires users to provide some type of authentication information, such as a password, PIN, or account number. The name of the user and the authentication information can be stored securely in a validation list. The ICS can use the information from the validation list rather than require all users of the ICS to have an iSeries user id and password.

An internet user can be permitted or denied access to the iSeries from the web server. The user, however, has no authority to any iSeries resources or authority to sign-on or run jobs. An iSeries user profile is never created for the internet users.

To create and delete validation lists, you can use the CL commands Create Validation List (CRTVLDL) and the Delete Validation List (DLTVLDL). Application Programming Interfaces (APIs) are also provided to allow applications to add, change, remove, verify (authenticate), and find entries in a validation list. For more information and examples, see the API topic in the Information Center (see"Prerequisite and related information" on page xvi for details).

Validation list objects are available for all applications to use. For example, if an application requires a password, the application passwords can be stored in a validation list object rather than a database file. The application can use the validation list APIs to verify a user's password, which is encrypted, rather than the application performing the verification itself.

In Version 4, Release 1, the authentication information (password, PIN, account number) that is associated with a validation list is always stored in a non–decryptable form, which cannot be returned to the user.

In Version 4, Release 2, you can choose to store the authentication information in a decryptable form. If a user has the appropriate security, the authentication information can be decrypted and returned to the user. For information about controlling the storage of decryptable data in validation lists, see "Retain Server Security (QRETSVRSEC)" on page 32.

Limit Access to Program Function

The limit access to program function allows you to define who can use an application, the parts of an application, or the functions within a program. This support is **not** a replacement for resource security. Limit access to program function does not prevent a user from accessing a resource (such as a file or program) from another interface.

The limit access to program function support provides APIs to:

- Register a function
- Retrieve information about the function
- Define who can or cannot use the function
- Check to see if the user is allowed to use the function

To use this support within an application, the application provider must register the functions when the application is installed. The registered function corresponds to a code block for specific functions in the application. When the user runs the application, the application calls the check usage API to see if the user is allowed to use the function that is associated with the code block, before invoking the code block. If the user is allowed to use the registered function, the code block is run. If the user is not allowed to use the function, the user is prevented from running the code block.

The system administrator specifies who is allowed or denied access to a function. The administrator can either use the API to manage the access to program function or use the iSeries Navigator. The OS/400 API section in the Information Center provides information about the limit access to program function APIs.

Chapter 8. Backup and Recovery of Security Information

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 *Backup and Recovery* book provides more information about backup and recovery. You may also refer to the Backup and Recovery topics in the iSeries Information Center (see"Prerequisite and related information" on page xvi for details).

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 115 shows the commands used to save and restore security information. The sections that follow discuss saving and restoring security information in more detail.

Table 115. 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	
Primary group [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	
Function registration information [2]		X		X	
Function usage information	X		X		X

[1] The SAVSECDTA, SAVSYS, and RSTUSRPRF commands save and restore ownership, primary group, primary group authority, and public authority for these object types : User profile (*USRPRF), Authorization list (*AUTL), and Authority holder (*AUTHLR).

[2] The object to save/restore is QUSEXRGOBJ, type *EXITRG in QUSRSYS library.

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
Primary group name
Primary group'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.

The uid and gid.

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.

Primary Group Information:

List of objects for which the profile is the primary group.

Auditing Information:

Action auditing value

Object auditing value

Function Usage Information:

Usage settings for registered functions.

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
Primary group name
Primary group's authority to object
Authorization list name
Field level authorities
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, owner, and primary group.

Authority Information Saved with User Profile:

The user profile attributes shown on the Create User Profile display.

Authority Table Saved Associated with User Profile:

One record for each private authority of the user profile, including usage settings for registered functions.

Function Registration Information Saved with QUSEXRGOBJ object:

The function registration information can be saved by saving the QUSEXRGOBJ *EXITRG object in QUSRSYS.

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 *Backup and Recovery* book 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), SECDTA(*PWDGRP) is not requested, 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)
  - Supplemental group profiles (SUPGRPPRF)

  Product passwords are changed to *NONE, so they will be incorrect after restoring an individual user profile that did not exist on the system.

- If profiles are being restored individually (RSTUSRPRF USRPRF(*ALL) is not specified) SECDTA(*PWDGRP) is not requested, and the profile exists on the system, the password, document password, and group profile are not changed.

  User profiles can be restored individually with the password and group information restored from the save media by specifying the SECDTA(*PWDGRP) parameter on the RSTUSRPRF command. *ALLOBJ and *SECADM special authorities are required to restore the password and group information when restoring individual profiles. Product passwords restored with the user profile will be incorrect after restoring an individual user profile that existed on the system, unless the SECDTA(*PWDGRP) parameter is specified on the RSTUSRPRF command.
- 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.

  Attention: User Profiles saved from a system with a different password level (QPWDLVL system value) than the system that is being restored may result in having a password that is not valid on the restored system. For example, if the saved user profile came from a system that was running password level 2, the user could have a password of ″This is my password″. This password would not be valid on a system running password level 0 or 1.

  Attention: 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 Service tools topic in the Information Center for instructions. See “Prerequisite and related information” on page xvi for more information about accessing the Information Center.
- If a profile exists on the system, the restore operation does not change the uid or gid.
- If a profile does not exist on the system, the uid and gid for a profile are restored from the save media. If either the uid or the gid already exists on the system, the system generates a new value and issues a message (CPI3810).
- *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 and the user performing the RSTUSRPRF does not have *ALLOBJ and *SECADM special authorities.
  - The profile was saved from the same system at security level 10 or 20.

  ATTENTION: 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 241 for additional considerations when restoring programs.

Primary group:

For an object that does not exist on the system:

- If the profile that is the primary group for the object is on the system, the primary group value and authority are restored for the object.
- If the profile that is the primary group does not exist on the system:
  - The primary group for the object is set to none.
  - The primary group authority is set to no authority.

When an existing object is restored, the primary group for the object is not changed by the restore operation.

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, the ALWOBJDIF parameter determines the result:
  - If ALWOBJDIF(*NONE) is specified, the existing object must have the same authorization list as the saved object. If not, the object is not restored.
  - If ALWOBJDIF(*ALL) is specified, the object is restored. The object is linked to the authorization list associated with the existing object.
- 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 241 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.

Function Registration Information:

- The function registration information can be restored by restoring the QUSEXRGOBJ *EXITRG object into QUSRSYS. This restores all of the registered functions. The usage information associated with the functions is restored when user profiles and authorities are restored.

Applications that Use Certificates Registration

- The applications that use certificates registration information can be restored by restoring the QUSEXRGOBJ *EXITRG object into QUSRSYS. This restores all of the registered applications. The association of the application to its certificate information can be restored by restoring the QYCDCERTI *USRIDX object into QUSRSYS.

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.

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 or the QSRSAVO API to find the profiles you want to restore.

Restoring Field Authority:

The following steps are required to restore private field authorities for database files that do not already exist on the system:

- Restore or create the necessary user profiles.
- Restore the files.
- Run the Restore Authority (RSTAUT) command.

The private field authorities are not fully restored until the private object authorities that they restrict are also established again.

Restoring Programs

Restoring programs to your system that are obtained from an unknown source poses a security exposure. Programs might perform operations that break your security requirements. Of particular concern are programs that contain restricted instructions, programs that adopt their owner authority, and programs that have been tampered with. This includes object types *PGM, *SRVPGM, *MODULE, and *CRQD. You can use the QVFYOBJRST, QFRCCVNRST, and QALWOBJRST system values to prevent these object types from being restored to your system. See Security-Related Restore System Values for more information about these system values.

The system uses a validation value to help protect programs. 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 the force conversion on restore (QFRCCVNRST) system value.

Note: Programs that are created for iSeries Version 5 Release 1 or later contain information that allows the program to be re-created at restore time if necessary. The information needed to re-create the program remains with the program even when the observability of the program is removed. If a program validation error is determined to exist at the time the program is

restored, the program will be re-created in order to correct the program validation error. The action of re-creating the program at restore time is not new to iseries Version 5 Release 1. In previous releases, any program validation error that was encountered at restore time resulted in the program being re-created if possible (if observability existed in the program being restored). The difference with iSeries Version 5 Release 1 or later programs is that the information needed to re-create the program remains even when observability was removed from the program.

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 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 238 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.

Attention: This procedure restores user profile values from the save media. See "Restoring User Profiles" on page 237 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.

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.

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 iSeries 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 iSeries system and accessing log and journal information on the system. You may want to create a special profile to be used by someone doing a security audit of your system. The auditor profile will need *AUDIT special authority to be able to change the audit characteristics of your system. Some of the auditing tasks suggested in this chapter require a user profile with *ALLOBJ and *SECADM special authority. Be sure that you set the password for the auditor profile to *NONE when the audit period has ended.

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 book. 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 book.

Physical Security

Note: The Basic System Security and Planning topic in the Information Center contains a complete discussion of physical security on the iSeries system. See "Prerequisite and related information" on page xvi for details.

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 Information Center for more information about the keylock switch (see "Prerequisite and related information" on page xvi for details).

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 40 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 password has been changed for the QSECOFR user profile. This profile is shipped with the password set to QSECOFR so you can sign on to install your system. The password **must** be changed the first time you sign-on your system and changed periodically after the installation.

Verify that it has been changed by checking a DSPAUTUSR list for the date the QSECOFR password was changed and by attempting to sign on with the default password.

Note: See "IBM-Supplied User Profiles" on page 116 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 userids and passwords

are changed, start DST and attempt to use the default values. See the topic "Working with service tools user IDs" on page 117 for more information.

Signing on with IBM-supplied user profiles, except QSECOFR, is not recommended. These IBM-supplied profiles are designed to own objects or to run system functions. Use a DSPAUTUSR list to verify that the following IBM-supplied user profiles have a password of *NONE:

QAUTPROF	QIPP	QSRV
QBRMS	QGATE	QSRVBAS
QCLUMGT	QLPAUTO	QSYS
QCLUSTER	QLPINSTALL	QSYSOPR
QCOLSRV	QMSF	QTCM
QDBSHR	QNETSPLF	QTCP
QDBSHRDO	QNFSANON	QTFTP
QDFTOWN	QNTP	QTMHHTP1
QDIRSRV	QPEX	QTMHHTTP
QDLFM	QPGMR	QTSTRQS
QDOC	QPM400	QUSER
QDSNX	QRJE	QYPSJSVR
QEJB	QSNADS	
QFNC	QSPL	
	QSPLJOB	

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 Security (GO SECURITY) 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 system 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. Use the DSPAUTUSR command to check for any group profiles that have passwords.

Whenever the system is not operating at password level 3 and users change their password, the system will attempt to create an equivalent password that is usable at the other password levels, if possible. You can use the DSPAUTUSR or PRTUSRPRF TYPE(*PWDINFO) commands to see which user profiles have passwords that are usable at the various password levels.

Note: The equivalent password is a best effort attempt to create a usable password for the other password levels but it may not have passed all of the password rules if the other password level was in effect. For example, if password BbAaA3x is specified at password level 2, the system will create an equivalent password of BBAAA3X for use at password levels 0 and 1. This would be true even if the QPWDLMTCHR system value includes 'A' as one of the limited characters (QPWDLMTCHR is not enforced at password level 2) or QPWDLMTREP system value specified that consecutive characters

cannot be the same (because the check is case sensitive at password level 2 but case insensitive at password levels 0 and 1).

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 at 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 "Printing Selected User Profiles" on page 278 shows how to 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 278 gives an example of how to determine this.

Programmers are restricted from production libraries. 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 231 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
```

You should use a naming convention for group profiles. When authorities are displayed, you can then easily recognize the group profile.

The administration of user profiles is adequately organized. No user profiles have large numbers of private authorities. The topic "Examining Large User Profiles" on page 278 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 only 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. You can use the DSPAUTUSR command for this information.

The password for an inactive employee is set to *NONE. Use the DSPAUTUSR command to verify that the inactive user profiles do not have passwords.

Management regularly verifies the users with special authorities, particularly *ALLOBJ *SAVSYS, and *AUDIT special authorities. The topic "Printing Selected User Profiles" on page 278 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) ASPDEV(*ALLAVL) 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.

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 "Library Lists" on page 193 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 279 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) when the auditing function is active. To audit authority failures, use the following system values and settings:

- QAUDCTL must be set to *AUDLVL
- QAUDLVL must include the values of *PGMFAIL and *AUTFAIL.

The best method to detect unauthorized attempts to access information is to review entries in the audit journal on a regular basis.

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.

Unauthorized Programs

The QALWOBJRST system value is set to *NONE to prevent anyone from restoring security-sensitive programs to the system.

The Check Object Integrity (CHKOBJITG) command is run periodically to detect unauthorized changes to program objects. This command is described in "Checking for Objects That Have Been Altered" on page 280.

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" on page 253 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.

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 E 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. The auditing of security-relevant events 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 116 on page 254 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.

Table 117 on page 255 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 model database outfile that can be used to define the record when you create an output file with the DSPJRN command. Complete layouts for the model database outfiles 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.

Table 116. 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 is 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.
*OPTICAL	Yes	Yes	**Optical functions**: All optical functions are audited, including functions related to optical files, optical directories, optical volumes, and optical cartridges. *OPTICAL may be used to detect attempts to create or delete an optical directory.
*NETCMN	Yes	No	**Network Communications Auditing**: The violations detected by the APPN Filter support are logged to the security auditing journal when the Directory search filter and the End point filter are audited.

Table 116. Action Auditing Values (continued)

Possible Value	Available on QAUDLVL System Value	Available on CHGUSRAUD Command	Description
*PGMADP	Yes	Yes	**Adopting authority**: The system writes a journal entry when adopted authority is used to 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, printing directly from a program, or sending a spooled file to a remote printer is logged. *PRTDTA may be used to detect printing confidential information.
*SAVRST	Yes	Yes	**Restore operations**: *SAVRST may be used to detect attempts to restore unauthorized 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 117. Security Auditing Journal Entries

Action or Object Auditing Value	Journal Entry Type	Model Database Outfile	Detailed Entry	Description
Action Auditing:				
*AUTFAIL [1]	AF	QASYAFJE/J4/J5	A	Attempt made to access an object or perform an operation to which the user was not authorized.
			F	ICAPI authorization error
			G	ICAPI authentication error
			J	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.

Table 117. Security Auditing Journal Entries (continued)

Action or Object Auditing Value	Journal Entry Type	Model Database Outfile	Detailed Entry	Description
			N	Profile token not a regenerable profile token
			P	Attempt made to use a profile handle that is not valid on the QWTSETP API.
			S	Attempt made to sign on without entering a user ID or a password.
			T	Not authorized to TCP/IP port
			U	A user permission request was not valid.
			V	Profile token not valid for generating new profile token
			W	Profile token not valid for swap
			X	Operation violation
			Y	Not authorized to the current JUID field during a clear JUID operation
			Z	Not authorized to the current JUID field during a set JUID operation
	AU	QASYAUJ5	E	Enterprise Identity Mapping (EIM) configuration change
	CV	QASYCVJ4/J5	E	Connection ended abnormally
	DI	QASYDIJ4/J5	AF	Authority failures
			PW	Password failures
			R	Connection rejected
	GR	QASYGRJ4/J5	F	Function registration operations.
	KF	QASYKFJ4/J5	P	An incorrect password was entered.
	IP	QASYIPJE/J4/J5	F	Authority failure for an IPC request.
	PW	QASYPWJE/J4/J5	A	APPC bind failure.
			D	An incorrect DST user name was entered.
			E	An incorrect DST password was entered.
			P	An incorrect password was entered.
			U	User name not valid
			X	Service tools user is disabled
			Y	Service tools user not valid
			Z	Service tools password not valid
	VO	QASYVOJ4/J5	U	Unsuccessful verify of a validation list entry.
	VC	QASYVCJE/J4/J5	R	A connection was rejected because of incorrect password.
	VN	QASYVNJE/J4/J5	R	A network logon was rejected because of expired account, incorrect hours, incorrect user id, or incorrect password.
	VP	QASYVPJE/J4/J5	P	An incorrect network password was used.
*CMD [2]	CD	QASYCDJE/J4/J5	C	A command was run.
			L	An S/36E control language statement was run.
			O	An S/36E operator control command was run.

Table 117. Security Auditing Journal Entries (continued)

Action or Object Auditing Value	Journal Entry Type	Model Database Outfile	Detailed Entry	Description
			P	An S/36E procedure was run.
			S	Command run after command substitution took place.
			U	An S/36E utility control statement was run.
*CREATE [3]	CO	QASYCOJE/J4/J5	N	Creation of a new object, except creation of objects in QTEMP library.
			R	Replacement of existing object.
	DI	QASYDIJ4/J5	CO	Object create
*DELETE [3]	DO	QASYDOJE/J4/J5	A	Object deleted
			C	Pending delete committed
			D	Pending create rolled back
			P	Delete pending
			R	Pending delete rolled back
	DI	QASYDIJ4/J5	DO	Object delete
*JOBDTA	JS	QASYJSJE/J4/J5	A	The ENDJOBABN command was used.
			B	A job was submitted.
			C	A job was changed.
			E	A job was ended.
			H	A job was held.
			I	A job was disconnected.
			M	Modify profile or group profile.
			N	The ENDJOB command was used.
			P	A program start request was attached to a prestart job.
			Q	Query attributes changed.
			R	A held job was released.
			S	A job was started.
			T	Modify profile or group profile using a profile token.
			U	CHGUSRTRC command.
	SG	QASYSGJE/J4/J5	A	Asynchronous AS/400 signal process.
			P	Asynchronous Private Address Space Environment (PASE) signal processed.
	VC	QASYVCJE/J4/J5	S	A connection was started.
			E	A connection was ended.
	VN	QASYVNJE/J4/J5	F	Logoff requested.
			O	Logon requested.
	VS	QASYVSJE/J4/J5	S	A server session was started.
			E	A server session was ended.
*NETCMN	CU	QASYCUJE/J4/J5	M	Creation of an object by the cluster control operation.
			R	Creation of an object by the Cluster Resource Group (*GRP) management operation.
	CV	QASYCVJ4/J5	C	Connection established.
			E	Connection ended normally.
	IR	QASYIRJ4/J5	L	IP rules have been loaded from from a file.
			N	IP rule have been unloaded for an IP Security connection.

Table 117. Security Auditing Journal Entries (continued)

Action or Object Auditing Value	Journal Entry Type	Model Database Outfile	Detailed Entry	Description
			P	IP rules have been loaded for and IP Security connection.
			R	IP rules have been read and copied to a file.
			U	IP rules have been unloaded (removed).
	IS	QASYISJ4/J5	1	Phase 1 negotiation.
			2	Phase 2 negotiation.
	ND	QASYNDJE/J4/J5	A	A violation was detected by the APPN Filter support when the Directory search filter was audited.
	NE	QASYNEJE/J4/J5	A	A violation is detected by the APPN Filter support when the End point filter is audited.
	SK	QASYSKJ4/J5	A	Accept
			C	Connect
			F	Filtered mail
			R	Reject mail
*OBJMGT [3]	DI	QASYDIJ4/J5	OM	Object rename
	OM	QASYOMJE/J4/J5	M	An object was moved to a different library.
			R	An object was renamed.
*OFCSRV	ML	QASYMLJE/J4/J5	O	A mail log was opened.
	SD	QASYSDJE/J4/J5	S	A change was made to the system distribution directory.
*OPTICAL	O1	QASY01JE/J4/J5	R	Open file or directory
			U	Change or retrieve attributes
			D	Delete file directory
			C	Create directory
			X	Release held optical file
	O2	QASY02JE/J4/J5	C	Copy file or directory
			R	Rename file
			B	Backup file or directory
			S	Save held optical file
			M	Move file
	O3	QASY03JE/J4/J5	I	Initialize volume
			B	Backup volume.
			N	Rename volume
			C	Convert backup volume to primary
			M	Import
			E	Export
			L	Change authorization list
			A	Change volume attributes
			R	Absolute read
*PGMADP	AP	QASYAPJE/J4/J5	S	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.

Table 117. Security Auditing Journal Entries (continued)

Action or Object Auditing Value	Journal Entry Type	Model Database Outfile	Detailed Entry	Description
			E	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.
			A	Adopted authority was used during program activation.
*PGMFAIL [1]	AF	QASYAFJE/J4/J5	B	A program ran a restricted machine interface instruction.
			C	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	A program accessed an object through an unsupported interface or callable program not listed as a callable API.
			E	Hardware storage protection violation.
			R	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/J4/J5	D	Printer output was printed directly to a printer.
			R	Output sent to remote system to print.
			S	Printer output was spooled and printed.
*SAVRST [3]	OR	QASYORJE/J4/J5	N	A new object was restored to the system.
			E	An object was restored that replaces an existing object.
	RA	QASYRAJE/J4/J5	A	The system changed the authority to an object being restored. [4]
	RJ	QASYRJJE/J4/J5	A	A job description that contains a user profile name was restored.
	RO	QASYROJE/J4/J5	A	The object owner was changed to QDFTOWN during restore operation.[4]
	RP	QASYRPJE/J4/J5	A	A program that adopts owner authority was restored.
	RQ	QASYRQJE/J4/J5	A	A *CRQD object with PROFILE(*OWNER) was restored.
	RU	QASYRUJE/J4/J5	A	Authority was restored for a user profile using the RSTAUT command.

Table 117. Security Auditing Journal Entries (continued)

Action or Object Auditing Value	Journal Entry Type	Model Database Outfile	Detailed Entry	Description
	RZ	QASYRZJE/J4/J5	A	The primary group for an object was changed during a restore operation.
*SECURITY	AD	QASYADJE/J4/J5	D	Auditing of a DLO was changed with CHGDLOAUD command.
			O	Auditing of an object was changed with CHGOBJAUD command.
			U	Auditing for a user was changed with CHGUSRAUD command.
	CA	QASYCAJE/J4/J5	A	Changes to authorization list or object authority.
	CP	QASYCPJE/J4/J5	A	Create, change, or restore operation of user profile.
	CQ	QASYCQJE/J4/J5	A	A *CRQD object was changed.
	CV	QASYCVJ4/J5	C	Connection established.
			E	Connection ended normally.
			R	Connection rejected.
	CY	QASYCYJ4/J5	A	Access Control function
			F	Facility Control function
			M	Master Key function
	DI	QASYDIJ4/J5	AD	Audit change
			BN	Successful bind
			CA	Authority change
			CP	Password change
			OW	Ownership change
			UB	Successful unbind
	DS	QASYDSJE/J4/J5	A	Request to reset DST QSECOFR password to system-supplied default.
			C	DST profile changed.
	EV	QASYEVJ4/J5	A	Add.
			C	Change.
			D	Delete.
	GR	QASYGRJ4/J5	A	Exit program added
			D	Edit program removed
			F	Function registration operation
			R	Exit program replaced
	GS	QASYGSJE/J4/J5	G	A socket descriptor was given to another job. (The GS audit record is created if it is not created for the current job.)
			R	Receive descriptor.
			U	Unable to use descriptor.
	IP	QASYIPJE/J4/J5	A	The ownership or authority of an IPC object was changed.
			C	Create an IPC object.
			D	Delete an IPC object.
			G	Get an IPC object.
	JD	QASYJDJE/J4/J5	A	The USER parameter of a job description was changed.
	KF	QASYKFJ4/J5	C	Certificate operation.
			K	Key ring file operation.
			T	Trusted root operation.
	NA	QASYNAJE/J4/J5	A	A network attribute was changed.

Table 117. Security Auditing Journal Entries (continued)

Action or Object Auditing Value	Journal Entry Type	Model Database Outfile	Detailed Entry	Description
	OW	QASYOWJE/J4/J5	A	object ownership was changed.
	PA	QASYPAJE/J4/J5	A	A program was changed to adopt owner authority.
	PG	QASYPGJE/J4/J5	A	The primary group for an object was changed.
	PS	QASYPSJE/J4/J5	A	A target user profile was changed during a pass-through session.
			E	An office user ended work on behalf of another user.
			H	A profile handle was generated through the QSYGETPH API.
			I	All profile tokens were invalidated.
			M	Maximum number of profile tokens have been generated.
			P	Profile token generated for user.
			R	All profile tokens for a user have been removed.
			S	An office user started work on behalf of another user.
			V	User profile authenticated.
	SE	QASYSEJE/J4/J5	A	A subsystem routing entry was changed.
	SO	QASYSOJ4/J5	A	Add entry.
			C	Change entry.
			R	Remove entry.
	SV	QASYSVJE/J4/J5	A	A system value was changed.
			B	Service attributes were changed.
			C	Change to system clock.
	VA	QASYVAJE/J4/J5	S	The access control list was changed successfully.
			F	The change of the access control list failed.
			V	Successful verify of a validation list entry.
	VU	QASYVUJE/J4/J5	G	A group record was changed.
			M	User profile global information changed.
			U	A user record was changed.
	X0	QASYX0J4/J5	1	Service ticket valid.
			2	Service principals do not match
			3	Client principals do not match
			4	Ticket IP address mismatch
			5	Decryption of the ticket failed
			6	Decryption of the authenticator failed
			7	Realm is not within client and local realms
			8	Ticket is a replay attempt
			9	Ticket not yet valid
			A	Deecrypt of KRB_AP_PRIV or KRB_AP_SAFE checksum error
			B	Remote IP address mismatch
			C	Local IP address mismatch

Table 117. Security Auditing Journal Entries (continued)

Action or Object Auditing Value	Journal Entry Type	Model Database Outfile	Detailed Entry	Description
			D	KRB_AP_PRIV or KRB_AP_SAFE timestamp error
			E	KRB_AP_PRIV or KRB_AP_SAFE replay error
			F	KRB_AP_PRIV KRB_AP_SAFE sequence order error
			K	GSS accept - expired credential
			L	GSS accept - checksum error
			M	GSS accept - channel bindings
			N	GSS unwrap or GSS verify expired context
			O	GSS unwrap or GSS verify decrypt/decode
			P	GSS unwrap or GSS verify checksum error
			Q	GSS unwrap or GSS verify sequence error
*SERVICE	ST	QASYSTJE/J4/J5	A	A service tool was used.
	VV	QASYVVJE/J4/J5	C	The service status was changed.
			E	The server was stopped.
			P	The server paused.
			R	The server was restarted.
			S	The server was started.
*SPLFDTA	SF	QASYSFJE/J4/J5	A	A spooled file was read by someone other than the owner.
			C	A spooled file was created.
			D	A spooled file was deleted.
			H	A spooled file was held.
			I	An inline file was created.
			R	A spooled file was released.
			U	A spooled file was changed.
*SYSMGT	DI	QASYDIJ4/J5	CF	Configuration changes
	SM	QASYSMJE/J4/J5	B	Backup options were changed using xxxxxxxxxx.
			C	Automatic cleanup options were changed using xxxxxxxxxx.
			D	A DRDA* change was made.
			F	An HFS file system was changed.
			N	A network file operation was performed.
			O	A backup list was changed using xxxxxxxxxx.
			P	The power on/off schedule was changed using xxxxxxxxxx.
			S	The system reply list was changed.
			T	The access path recovery times were changed.
	VL	QASYVLJE/J4/J5	A	The account is expired.
			D	The account is disabled.
			L	Logon hours were exceeded.
			U	Unknown or unavailable.
			W	Workstation not valid.

Table 117. Security Auditing Journal Entries (continued)

Action or Object Auditing Value	Journal Entry Type	Model Database Outfile	Detailed Entry	Description
Object Auditing:				
*CHANGE	DI	QASYDIJ4/J5	IM	LDAP directory import
			ZC	Object changes
	GR	QASYGRJ4/J5	F	Function registration operations[6]
	LD	QASYLDJE/J4/J5	L	Link a directory.
			U	Unlink a directory.
			K	Search a directory.
	VF	QASYVFJE/J4/J5	A	The file was closed because of administrative disconnection.
			N	The file was closed because of normal client disconnection.
			S	The file was closed because of session disconnection.
	VO	QASYVOJ4/J5	A	Add validation list entry.
			C	Change validation list entry.
			F	Find validation list entry.
			R	Remove validation list entry.
	VR	QASYVRJE/J4/J5	F	Resource access failed.
			S	Resource access was successful.
	YC	QASYYCJE/J4/J5	C	A document library object was changed.
	ZC	QASYZCJE/J4/J5	C	An object was changed.
*ALL [5]	DI	QASYDIJ4/J5	EX	LDAP directory export
			ZR	Object read
	GR	QASYGRJ4/J5	F	Function registration operations[6]
	YR	QASYYRJE/J4/J5	R	A document library object was read.
	ZR	QASYZRJE/J4/J5	R	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 238 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 (DI, YC, YR, ZC, ZR) are written.

[6] When the QUSRSYS/QUSEXRGOBJ *EXITRG object is being audited.

Planning the Auditing of Object Access

The system provides the ability to log accesses to an object in the security audit journal. This is called **object auditing**. 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 118 on page 264 shows how the OBJAUD values for the object and the user profile work together.

Table 118. 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
ASP device . . . . . . . . . . .   *
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 274.
- 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 274.

- 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 . . . . . . . . . . . .  *ALLAVL
Object type  . . . . . . . . . .   *ALL
ASP device . . . . . . . . . . .   *
Object auditing value  . . . . .   *USRPRF
```

     Attention: 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 and directories 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.

Choosing to not audit QTEMP objects

The value, *NOQTEMP, can be specified as a value for system value QAUDCTL. If specified, you must also specify either *OBJAUD or *AUDLVL. When auditing is active and *NOQTEMP is specified the following actions on objects in the QTEMP library will NOT be audited.

Changing or reading objects in QTEMP (journal entry types ZC, ZR).

Changing the authority, owner, or primary group of objects in QTEMP (journal entry types CA, OW, PG).

Using CHGSECAUD to Set up Security Auditing

Overview:

Purpose:

Set up the system to collect security events in the QAUDJRN journal.

How To:

CHGSECAUD
DSPSECAUD

Authority:

The user must have *ALLOBJ and *AUDIT special authority.

Journal Entry:

CO (create object)
SV (system value change)
AD (object and user audit changes)

Notes: The CHGSECAUD command creates the journal and journal receiver if it does not exist. The CHGSECAUD then sets the QAUDCTL and QAUDLVL system values.

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) +
           THRESHOLD(100000) AUT(*EXCLUDE)   +
           TEXT('Auditing Journal Receiver')
```

   - Place the journal receiver in a library that is saved regularly. Do **not** place the journal receiver in library QSYS, even though that is where the journal will be.
   - Choose a journal receiver name that 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. Using this type of naming convention is also useful if you choose to have the system manage changing your journal receivers.
   - Specify a receiver threshold appropriate to your system size and activity. The size you choose should be based on the number of transactions on your system and the number of actions you choose to audit. If you use system change-journal management support, the journal receiver threshold must be at least 5,000KB. For more information on journal receiver threshold refer to the *Backup and Recovery* book.
   - 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) +
        MNGRCV(*SYSTEM) DLTRCV(*NO) +
        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.
   - Use the *Manage receiver* (MNGRCV) parameter to have the system change the journal receiver and attach a new one when the attached receiver exceeds the threshold specified when the journal receiver was created. If you choose this option, you do not have to use the CHGJRN command to detach receivers and create and attach new receivers manually.
   - Do not have the system delete detached receivers. Specify DLTRCV(*NO), which is the default. The QAUDJRN receivers are your security audit trail. Ensure that they are adequately saved before deleting them from the system.

   The *Backup and Recovery* book 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 253.
4. Set action auditing for individual users if necessary using the CHGUSRAUD command. See "Planning the Auditing of Actions" on page 253.
5. Set object auditing for specific objects if necessary using the CHGOBJAUD and CHGDLOAUD commands. See "Planning the Auditing of Object Access" on page 263.
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 266.
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 265.
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. Objects should not be journaled to the audit journal. Commitment control should not use the audit journal. User entries should not be sent to this journal using the Send Journal Entry (SNDJRNE) command or the Send Journal Entry (QJOSJRNE) API.

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
- ENDJRNxxx (End Journaling) commands
- APYJRNCHG command
- RMVJRNCHG command
- DMPOBJ or DMPSYSOBJ command
- Moving the journal
- 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.

You may want to have the system manage the changing of journal receivers. Specify MNGRCV(*SYSTEM) when you create the QAUDJRN journal, or change the journal to that value. If you specify MNGRCV(*SYSTEM), the system automatically detaches the receiver when it reaches its threshold size and creates and attaches a new journal receiver. This is called **system change-journal management**.

If you specify MNGRCV(*USER) for the QAUDJRN, a message is sent to the threshold message queue specified for the journal when the journal receiver reaches a storage threshold. The message indicates that the receiver has reached its threshold. Use the CHGJRN command to detach the receiver and attach a new journal receiver. This prevents *Entry not journaled* error conditions. If you do receive a message, you must use the CHGJRN command for security auditing to continue.

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 as 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 (CPF7099), you can automatically attach a new receiver. If you use system change-journal management, then message CPF7020 is sent to the journal message queue when a system change journal is completed. You can monitor for this message to know when to do a save of the detached journal receivers.

Attention: 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 *Backup and Recovery* book 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 attach 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.

Having the system manage receivers automatically is the recommended approach. You can specify this by using the *Manage receiver* parameter when you create the journal.

If you have set up action auditing and object auditing to log many different events, you may need to specify a large threshold value for the journal receiver. If you are managing receivers manually, 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.

System-Managed Journal Receivers: If you have the system manage the receivers, use the following procedure to save all detached QAUDJRN receivers and to delete them:

1. Type `WRKJRNA QAUDJRN`. The display shows you the currently attached receiver. Do not save or delete this receiver.
2. Use F15 to work with the receiver directory. This shows all receivers that have been associated with the journal and their status.
3. Use the SAVOBJ command to save each receiver, except the currently attached receiver, which has not already been saved.
4. Use the DLTJRNRCV command to delete each receiver after it is saved.

Note: An alternative to the above procedure could be done using the journal message queue and monitoring for the CPF7020 message which indicates that the system change journal has completed successfully. See the *Backup and Recovery* for more information on this support.

User-Managed Journal Receivers: If you choose to manage journal receivers manually, 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, the system creates and attaches a new receiver called AUDRCV0004.

   The Work with Journal Attributes (WRKJRNA) command tells you which receiver is currently attached: `WRKJRNA QAUDJRN`.

2. Use the Save Object (SAVOBJ) command to save the detached 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 detached receiver, using the SAVOBJ and DLTJRNRCV commands.
4. You can delete the QAUDJRN journal once you change QAUDCTL to *NONE. If you plan to resume security auditing in the future, you may want to leave the QAUDJRN journal on the system. However, if the QAUDJRN journal is set up with MNGRCV(*SYSTEM), the system detaches the receiver and attaches a new one whenever you perform an IPL, whether or not security auditing is active. You need to delete these journal receivers. Saving them before deleting them should not be necessary, because they do not contain any audit entries.

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 Journal Entries (DSPAUDJRNE) 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).

The default is to display entries from only the attached receiver. You can use RCVRNG(*CURCHAIN) to see entries from all receivers that are in the receiver chain for the QAUDJRN journal, up to and including the receiver that is currently attached.

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                          JONES1      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                          SMITHJ      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

Object . . . . . :   QAUDJRN         Library  . . . . . :   QSYS
Member . . . . . :                   Sequence . . . . . :   28026
Code . . . . . :   T  - Audit trail entry
Type . . . . . :   PW - Invalid password or user ID

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 additional parameters prompting you for information about the output file:

```
                        Display Journal (DSPJRN)

Type choices, press Enter.
   :
Output . . . . . . . . . . . . . > *OUTFILE
Outfile format . . . . . . . . .   *TYPE4
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, such as the entry type, the date of the entry, and the job that caused the entry. The QJORDJE4 record format is provided to define these fields when you specify *TYPE4 as the outfile format parameter. See Table 142 on page 503 for more information.

For more information on other records and their outfile formats see Appendix F.

If you want to perform a detailed analysis of a particular entry type, use one of the model database outfiles 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(QASYAFJ4) 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(*TYPE4) OUTFILE(QGPL/AUDJRNAF)
```

3. Use Query or a program to analyze the information in the AUDJRNAF file.

Table 117 on page 255 shows the name of the model database outfile for each entry type. Appendix F shows the file layouts for each model database outfile.

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 QASYPWJ4 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 QASYAFJ4 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 to determine what 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 QASYAFJ4 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 117 on page 255 shows which journal entry is written for each authority violation message.

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.

You will find additional information in Appendix G, "Commands and Menus for Security Commands" on page 599. This appendix includes examples to use the commands and information about the menus for the security tools.

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. The QHST log should not be considered a complete source of security violations. Use the security audit functions instead.

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.

Using Journals to Monitor Object 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 objects, 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 object was accessed. It does not log every transaction to the object. For critical objects on your system, you may want more detailed information about the specific data that was accessed and changed. Object journaling is used primarily for object integrity and recovery. Refer to the *Backup and Recovery* book for a list of object types which can be journaled, and what is journaled for each object type. A security officer or auditor can also use these journal entries to review object changes. Do not journal any objects to the QAUDJRN journal.

Journal entries can include:

- Identification of the job and user and the time of access
- Before- and after-images of all object changes
- Records of when the object was opened, closed, changed, saved, etc.

A journal entry cannot be altered by any user, even the security officer. A complete journal or journal receiver can be deleted, but this is easily detected.

If you are journaling files 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 are journaling other object types and want to see the information for a particular object, type the following:

```
DSPJRN JRN(library/journal)
       OUTPUT(*OUTFILE)
       OUTFILEFMT(*TYPE4)
       OUTFILE(library/outfile)
       ENTDTALEN(*CALC)
```

You can then do a query or use SQL to select all of the records from this outfile for a specific object name.

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 objects are being journaled by a particular journal, use the Work with Journal Attributes (WRKJRNA) command.

The *Backup and Recovery* book 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    08/04/0x            Roger Anders
            VINCENTM    09/15/0x            Mark Vincent
DPTWH
            ANDERSOR    08/04/0x            Roger Anders
            WAGNERR     09/06/0x            Rose Wagner
QSECOFR
            JONESS      09/20/0x            Sharon Jones
            HARRISOK    08/29/0x            Ken Harrison
*NO GROUP
            DPTSM       09/05/0x    X       Sales and Marketing
            DPTWH       08/13/0x    X       Warehouse
            RICHARDS    09/05/0x            Janet Richards
            SMITHJ      09/18/0x            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 a query tool.

```
DSPUSRPRF USRPRF(*ALL) +
          TYPE(*BASIC) OUTPUT(*OUTFILE)
```

You can use a query tool to create 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 model database outfile) 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.
- List all users who do not have a password for use at password levels 0 and 1 by selecting records where the Password present for level 0 or 1 field (called UPENPW in the model outfile) is equal to N.
- List all users who have a password for use at password levels 2 and 3 by selecting records where the Password present for level 2 or 3 field (called UPENPH in the model outfile) is equal to Y.

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(QSYS/*ALL) OBJTYPE(*LIB) ASPDEV(*ALLAVL) OUTPUT(*PRINT)
   ```

2. Use the Display Object Authority (DSPOBJAUT) command to list the authorities to a specific library:

   ```
   DSPOBJAUT OBJ(library-name) OBJTYPE(*LIB) +
             ASPDEV(asp-device-name) OUTPUT(*PRINT)
   ```

3. Use the Display Library (DSPLIB) command to list the objects in the library:

   ```
   DSPLIB LIB(library-name) ASPDEV(asp-device-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" on page 278 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) ASPDEV(asp-device-name) 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) ASPDEV(asp-device-name) DETAIL(*FULL)
```

Checking for Objects That Have Been Altered

You can use the Check Object Integrity (CHKOBJITG) command to look for objects that have been altered. An altered object is usually an indication that someone is attempting to tamper with your system. You may want to run this command after someone has:

- Restored programs to your system
- Used dedicated service tools (DST)

When you run the command, the system creates a database file containing information about any potential integrity problems. You can check objects owned by one or more profiles, objects that match a path name, or all objects on the system. You can look for objects whose domain has been altered and objects that have been tampered with. You can recalculate program validation values to look for objects of type *PGM, *SRVPGM, *MODULE, and *SQLPKG that have been altered. You can check the signature of objects that can be digitally signed. You can also check if libraries and commands have been tampered with.

Running the CHKOBJITG program requires *AUDIT special authority. The command may take a long time to run because of the scans and calculations it performs. You should run it at a time when your system is not busy. Most IBM commands duplicated from a release prior to V5R2 will be logged as violations. These commands should be deleted and re-created using the CRTDUPOBJ (Create duplicate object) command each time a new release is loaded.

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 116 on page 254 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 274.

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 topic in the Information Center contains more detailed information about these commands. See "Prerequisite and related information" on page xvi for details. The tables in Appendix D show what object authorities are required to use these commands.

Table 119. 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 120. 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 121. Commands for Working with Object Authority and Auditing

Command Name	Descriptive Name	Function
CHGAUD	Change Auditing	Allows you to change the auditing value for an object.
CHGAUT	Change Authority	Allows you to change the authority of users to objects.
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.
CHGOBJPGP	Change Object Primary Group	Allows you to change the primary group for an object to another user or to no primary group.
CHGOWN	Change Owner	Allows you to change the ownership of an object from one user to another.
CHGPGP	Change Primary Group	Allows you to change the primary group for an object to another user or to no primary group.
DSPAUT	Display Authority	Allows you to display users' authority to an object.
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.
WRKAUT	Work with Authority	Allows you to work with object authority by selecting options on a list display.
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.
WRKOBJPGP	Work with Objects by Primary Group	Allows you to work with the objects for which a profile is the primary group using options from a list display.

Table 122. Commands for Working with Passwords

Command Name	Descriptive Name	Function
CHGDSTPWD	Change Dedicated Service Tools Password	Allows you to reset the DST security capabilites profile 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.
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.
CRTUSRPRF[1]	Create User Profile	When you add a user to the system, you assign a password to the user.

[1] When a CRTUSRPRF is done, you can't specify that the *USRPRF is to be created into an IASP. However, when a user is privately authorized to an object on and IASP, is the owner of an object on an IASP, or is the primary group of an object on an IASP, the profile's name is stored on the IASP. If the IASP is moved to another system, the private authority, object ownership, and primary group entries will be attached to the profile with the same name on the target system. If a profile does not exist on the target system, a profile will be created. The user will not have any special authorities and the password will be set to *NONE.

Table 123. 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.
CHKOBJITG	Check Object Integrity	Check the objects owned by one or more user profiles or check the objects that match the pathname to ensure the objects have not been tampered with.
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 usable at any password level, whether the user profile has a password usable at the various password levels, whether the user profile has a password usable with NetServer, 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.
PRTPRFINT	Print Pofile Internals	Allows you to print a report of internal information on the number of entries.
PRTUSRPRF	Print User Profile	Allows you to analyze user profiles that meet specified criteria.
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 124. 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 125. Commands for Working with Auditing

Command Name	Descriptive Name	Function
CHGAUD	Change 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.
CHGOBJAUD	Change Object Auditing	Allows you to specify the auditing for an object.
CHGUSRAUD	Change User Audit	Allows you to specify the action and object auditing for a user profile.

Table 126. 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.
CHGDLOPGP	Change Document Library Object Primary Group	Allows you to change the primary group for a document library object.
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.

Table 126. Commands for Working with Document Library Objects (continued).

Command Name	Descriptive Name	Function
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.

Table 127. Commands for Working with Server Authentication Entries

Command Name	Descriptive Name	Function
ADDSVRAUTE	Add Server Authentication Entry	Allows you to add server authentication information for a user profile.
CHGSVRAUTE	Change Server Authentication Entry	Allows you to change existing server authentication entries for a user profile.
DSPSVRAUTE	Display Server Authentication Entries	Allows you to display server authentication entries for a user profile.
RMVSVRAUTE	Remove Server Authentication Entry	Allows you to remove server authentication entries from the specified user profile.

These commands allow a user to specify a user name, the associated password, and the name of a remote server machine. Distributed Relational Database Access (DRDA) uses these entries to run database access requests as the specified user on the remote server.

Table 128. 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.
WRKDIRE	Work with Directory	Provides a set of displays that allow a user to view, add, change, and remove entries in the system distribution directory.

Table 129. Commands for Working with Validation Lists

Command Name	Descriptive Name	Function
CRTVLDL	Create Validation List	Allows you to create a validation list object that contains entries consisting of an identifier, data that will be encrypted by the system when it is stored, and free-form data.
DLTVLDL	Delete Validation List	Allows you to delete the specified validation list from a library.

The following tables describe several different kinds of security tools. For more information on the security tools, see Appendix G, "Commands and Menus for Security Commands".

Table 130. Security Tools for Working with Auditing

Command Name	Descriptive Name	Function
CHGSECAUD	Change Security Auditing	Allows you to set up security auditing and to change the system values that control security auditing.
DSPAUDJRNE	Display Audit Journal Entries	Allows you to display or print information about entries in the security audit journal. You can select specific entry types, specific users, and a time period.
DSPSECAUD	Display Security Auditing Values	Allows you to display information about the security audit journal and the system values that control security auditing.

Table 131. Security Tools for Working with Authorities

Command Name	Descriptive Name	Function
PRTJOBDAUT	Print Job Description Authority	Allows you to print a list of job descriptions whose public authority is not *EXCLUDE. You can use this command to print information about job descriptions that specify a user profile that every user on the system can access.
PRTPUBAUT	Print Publicly Authorized Objects	Allows you to print a list of objects of the specified type whose public authority is not *EXCLUDE.
PRTPVTAUT	Print Private Authorities	Allows you to print a list of private authorities for objects of the specified type.
PRTQAUT	Print Queue Authority	Allows you to print the security settings for output queues and job queues on your system. These settings control who can view and change entries in the output queue or job queue.
PRTSBSDAUT	Print Subsystem Description Authority	Allows you to print a list of subsystem descriptions in a library that contains a default user in a subsystem entry.
PRTTRGPGM	Print Trigger Programs	Allows you to print a list of trigger programs that are associated with database files on your system.
PRTUSROBJ	Print User Objects	Allows you to print a list of the user objects (objects not supplied by IBM) that are in a library.

Table 132. Security Tools for Working with System Security

Command Name	Descriptive Name	Function
CHGSECA [1]	Change Security Attributes	Allows you to set new starting values for generating user ID numbers or group ID numbers. Users can specify a starting user ID number and a starting group ID number.
CFGSYSSEC	Configure System Security	Allows you to set security-relevant system values to their recommended settings. The command also sets up security auditing on your system.
CLRSVRSEC	Clear Server Security Data	Allows you to clear decryptable authentication information that is associated with user profiles and validation list (*VLDL) entries. **Note:** This is the same information that was cleared in releases previous to V5R2 when the QRETSVRSEC system value was changed from '1' to '0'.
DSPSECA	Display Security Attributes	Allows you to display the current and pending values of some system security attributes.
PRTCMNSEC	Print Communications Security	Allows you to print the security attributes of the *DEVD, *CTL, and *LIND objects on the system.
PRTSYSSECA	Print System Security Attributes	Allows you to print a list of security-relevant system values and network attributes. The report shows the current value and the recommended value.
RVKPUBAUT	Revoke Public Authority	Allows you to set the public authority to *EXCLUDE for a set of security-sensitive commands on your system.

1 To use this command, you must have *SECADM special authority.

For more information on tools and suggestions about how to use the security tools, see the *Tips for Making Your iSeries 400 Secure* book, GC41-0615.

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 133 shows the default values that are used for all IBM-supplied user profiles and on the Create User Profile (CRTUSRPRF) command. The parameters are sequenced in the order they appear on the Create User Profile display.

Table 134 lists each IBM-supplied profile, its purpose, and any values for the profile that are different from the defaults for IBM-supplied user profiles.

Note:

> Table 134 now includes additional user profiles that are shipped with the licensed program products. The table includes only **some**, but not all user profiles for licensed program products; therefore, the list is not inclusive.

Attention:

- Password for the QSECOFR profile

  You **must change** the password for the QSECOFR profile after you install your system. This password is the same for every iSeries system and poses a security exposure until it is changed. However, do **not** change any other values for IBM-supplied user profiles. Changing these profiles may cause system functions to fail.
- Authorities for IBM-supplied profies

  Use **caution** when removing authorities that IBM-supplied profiles have to objects that are shipped with the operating system. Some IBM-supplied profiles are granted private authorities to objects that are shipped with the operating system. Removing any of these authorities may cause system functions to fail.

Table 133. Default Values for User Profiles

	Default Values	
User Profile Parameter	**IBM-Supplied User Profiles**	**Create User Profile Display**
Password (PASSWORD)	*NONE	*USRPRF[4]
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)	*SYSVAL	*SYSVAL
Display sign-on information (DSPSGNINF)	*SYSVAL	*SYSVAL

Table 133. Default Values for User Profiles (continued)

User Profile Parameter	Default Values	
	IBM-Supplied User Profiles	**Create User Profile Display**
Password expiration interval (PWDEXPITV)	*SYSVAL	*SYSVAL
Limit device sessions (LMTDEVSSN)	*SYSVAL	*SYSVAL
Keyboard buffering (KBDBUF)	*SYSVAL	*SYSVAL
Maximum storage (MAXSTG)	*NOMAX	*NOMAX
Priority limit (PTYLMT)	0	3
Job description (JOBD)	QDFTJOBD	QDFTJOBD
Job description library	QGPL	*LIBL
Group profile (GRPPRF)	*NONE	*NONE
Owner (OWNER)	*USRPRF	*USRPRF
Group authority (GRPAUT)	*NONE	*NONE
Group authority type (GRPAUTTYP)	*PRIVATE	*PRIVATE
Supplemental groups (SUPGRPPRF)	*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 (PRTDEV)	*WRKSTN	*WRKSTN
Output queue (OUTQ)	*WRKSTN	*WRKSTN
Attention program (ATNPGM)	*NONE	*SYSVAL
Sort sequence (SRTSEQ)	*SYSVAL	*SYSVAL
Language identifier (LANGID)	*SYSVAL	*SYSVAL
Country or Region Identifier (CNTRYID)	*SYSVAL	*SYSVAL
Coded Character Set Identifier (CCSID)	*SYSVAL	*SYSVAL
Set Job Attributes (SETJOBATR)	*SYSVAL	*SYSVAL
Locale (LOCALE)	*NONE	*SYSVAL
User Option (USROPT)	*NONE	*NONE
User Identification Number (UID)	*GEN	*GEN
Group Identification Number (GID)	*NONE	*NONE
Home Directory (HOMEDIR)	*USRPRF	*USRPRF
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.

[4] When you perform a CRTUSRPRF, you can not create a user profile (*USRPRF) into an independent disk pool. However, when a user is privately authorized to an object in the independent disk pool, is the owner of an object on an independent disk pool, or is the primary group of an object on an independent disk pool, the name of the profile is stored on the independent disk pool. If the independent disk pool is moved to another system, the private authority, object ownership, and primary group entries will be attached to the profile with the same name on the target system. If a profile does not exist on the target system, a profile will be created. The user will not have any special authorities and the password will be set to *NONE.

Table 134. IBM-Supplied User Profiles

Profile Name	Descriptive Name	Parameters Different from Default Values
QADSM	ADSM user profile	• USERCLS: *SYSOPR • CURLIB: QADSM • TEXT: ADSM profile used by ADSM server • SPCAUT: *JOBCTL, *SAVSYS • JOBD: QADSM/QADSM • OUTQ: QADSM/QADSM
QAFOWN	APD user profile	• USRCLS: *PGMR • SPCAUT: *JOBCTL • JOBD: QADSM/QADSM • TEXT: Internal APD User Profile
QAFUSR	APD user profile	• TEXT: Internal APD User Profile
QAFDFTUSR	APD user profile	• INLPGM: *LIBL/QAFINLPG • LMTCPB: *YES • TEXT: Internal APD User Profile
QAUTPROF	IBM authority user profile	
QBRMS	BRM user profile	
QCLUMGT	Cluster management profile	• STATUS: *DISABLED • MSGQ: *NONE • ATNPGM: *NONE
QCLUSTER	High availability cluster profile	• SPCAUT: *IOSYSCFG
QCOLSRV	Management central collection services user profile	
QDBSHR	Database share profile	• AUT: *ADD, *DELETE
QDBSHRDO	Database share profile	• AUT: *ADD, *DELETE
QDCEADM	DCE user profile	• PASSWORD: *USRPRF • PWDEXP: *YES • STATUS: *DISABLED • TEXT: *NONE • SPCAUT: *JOBCTL
QDFTOWN	Default owner profile	• PTYLMT: 3 • ACGCDE: *BLANK
QDIRSRV	OS/400 Directory services server user profile	• LMTCPB: *YES • JOBD: QGPL/QBATCH • DSPSGNINF: *NO • LMTDEVSSN: *NO • DLVRY: *HOLD • SPCENV: *NONE • ATNPGM: *NONE

Table 134. IBM-Supplied User Profiles (continued)

Profile Name	Descriptive Name	Parameters Different from Default Values
QDLFM	DataLink File Manager profile	• SRTSEQ: *HEX
QDOC	Document profile	• ACGCDE: *BLANK • AUT: *CHANGE
QDSNX	Distributed systems node executive profile	• PTYLMT: 3 • CCSID: *HEX • ACGCDE: *BLANK • SRTSEQ: *HEX
QEJB	Enterprise Java user profile	
QFNC	Finance profile	• PTYLMT: 3
QGATE	VM/MVS* bridge profile	• CCSID: *HEX • SRTSEQ: *HEX
QIPP	Internet printing profile	• MSGQ: QUSRSYS/QIPP
QLPAUTO	Licensed program automatic install profile	• USRCLS: *SYSOPR • INLMNU: *SIGNOFF • SPCAUT: *ALLOBJ, *JOBCTL ,*SAVSYS, *SECADM, *IOSYSCFG • INLPGM: QLPINATO • Library: QSYS • DLVRY: *HOLD • SEV: 99
QLPINSTALL	Licensed program install profile	• USRCLS: *SYSOPR • DLVRY: *HOLD • SPCAUT: *ALLOBJ, *JOBCTL, *SAVSYS, *SECADM, *IOSYSCFG
QMSF	Mail server framework profile	• CCSID: *HEX • SRTSEQ: *HEX
QMQM	MQSeries® user profile	• USRCLS: *SECADM • SPCAUT: *NONE • PRTDEV: *SYSVAL • TEXT: MQM user which owns the QMQM library
QNFSANON	NFS user profile	
QNETSPLF	Network spooling profile	
QNETWARE	ECS user profile	• STATUS: *DISABLED • TEXT: QFPNTWE USER PROFILE
QNTP	Network time profile	• JOBD: QTOTNTP • JOBD LIBRARY: QSYS

Table 134. IBM-Supplied User Profiles (continued)

Profile Name	Descriptive Name	Parameters Different from Default Values
QOIUSER	OSI Communication Subsystem	• USRCLS: *SYSOPR • SPCAUT: *JOBCTL, *SAVSYS, *IOSYSCFG • CURLIB: QOSI • MSGQ: QOSI/QOIUSER • DLVRY: *HOLD • OUTQ: *DEV • PRTDEV: *SYSVAL • ATNPGM: *NONE • CCSID: *HEX • TEXT: Internal OSI Communication Subsystem User Profile
QOSIFS	OSI File Server User Profile	• USRCLS: *SYSOPR • SPCAUT: *JOBCTL, *SAVSYS • OUTQ: *DEV • CURLIB: *QOSIFS • CCSID: *HEX • TEXT: Internal OSI File Services User Profile
QPGMR	Programmer profile	• USRCLS: *PGMR • SPCAUT: *ALLOBJ [1] *SAVSYS *JOBCTL • PTYLMT: 3 • ACGCDE: *BLANK
QPEX	Performance Explorer user profile	• PTYLMT: 3 • ATNPGM: *SYSVAL • TEXT: IBM-supplied User Profile • ACGCDE: *BLANK
QPM400	Performance Management/400 (PM/400)	• SPCAUT: *IOSYSCFG, *JOBCTL
QPRJOWN	Parts and projects owner user profile	• STATUS: *DISABLED • CURLIB: QADM • TEXT: User profile of parts and projects owner
QRDARSADM	R/DARS user profile	• INLMNU: *SIGNOFF • TEXT: R/DARS Administration Profile
QRDAR	R/DARS owning profile	• USRCLS: *PGMR • INLMNU: *SIGNOFF • OUTQ: *DEV • TEXT: R/DARS-400 owning profile
QRDARS4001	R/DARS owning profile 1	• INLMNU: *SIGNOFF • GRPPRF: QRDARS400 • OUTQ: *DEV • TEXT: R/DARS-400 owning profile 1

Table 134. IBM-Supplied User Profiles (continued)

Profile Name	Descriptive Name	Parameters Different from Default Values
QRDARS4002	R/DARS owning profile 2	• INLMNU: *SIGNOFF • GRPPRF: QRDARS400 • OUTQ: *DEV • TEXT: R/DARS-400 owning profile 2
QRDARS4003	R/DARS owning profile 3	• INLMNU: *SIGNOFF • GRPPRF: QRDARS400 • OUTQ: *DEV • TEXT: R/DARS-400 owning profile 3
QRDARS4004	R/DARS owning profile 4	• INLMNU: *SIGNOFF • GRPPRF: QRDARS400 • OUTQ: *DEV • TEXT: R/DARS-400 owning profile 4
QRDARS4005	R/DARS owning profile 5	• INLMNU: *SIGNOFF • GRPPRF: QRDARS400 • OUTQ: *DEV • TEXT: R/DARS-400 owning profile 5
QRMTCAL	Remote Calendar user profile	• TEXT: OfficeVision® Remote Calendar User
QRJE	Remote job entry profile	• USRCLS: *PGMR • SPCAUT: *ALLOBJ [1] *SAVSYS [1] *JOBCTL
QSECOFR	Security officer profile	• PWDEXP: *YES • USRCLS: *SECOFR • SPCAUT: *ALLOBJ, *SAVSYS, *JOBCTL, *SECADM, *SPLCTL, *SERVICE, *AUDIT, *IOSYSCFG • ACGCDE: *BLANK • UID: 0 • PASSWORD: QSECOFR
QSNADS	SNA distribution services profile	• CCSID: *HEX • SRTSEQ: *HEX
QSOC	OptiConnect user profile	• USRCLS: *SYSOPR • CURLIB: *QSOC • SPCAUT: *JOBCTL • MSGQ: QUSRSYS/QSOC
QSPL	Spool profile	
QSPLJOB	Spool job profile	• AUT: *USE
QSRV	Service profile	• USRCLS: *PGMR • SPCAUT: *ALLOBJ [1], *SAVSYS [1], *JOBCTL, *SERVICE • ASTLVL: *INTERMED • ATNPGM: QSCATTN • Library: QSYS

Table 134. IBM-Supplied User Profiles (continued)

Profile Name	Descriptive Name	Parameters Different from Default Values
QSRVBAS	Service basic profile	• USRCLS: *PGMR • SPCAUT: *ALLOBJ [1] *SAVSYS [1] *JOBCTL • ASTLVL: *INTERMED • ATNPGM: QSCATTN • Library: QSYS
QSVCCS	CC Server user profile	• USRCLS: *SYSOPR • SPCAUT: *JOBCTL • SPCENV: *SYSVAL • TEXT: CC Server User Profile
QSVCM	Client Management Server user profile	• TEXT: Client Management Server User Profile
QSVSM	ECS user profile	• USRCLS: *SYSOPR • STATUS: *DISABLED • SPCAUT: *JOBCTL • SPCENV: *SYSVAL • TEXT: SystemView® System Manager User Profile
QSVSMSS	Managed System Service user profile	• STATUS: *DISABLED • USRCLS: *SYSOPR • SPCAUT: *JOBCTL • SPCENV: *SYSVAL • TEXT: Managed System Service User Profile
QSYS	System profile	• USRCLS: *SECOFR • SPCAUT: *ALLOBJ, *SECADM, *SAVSYS, *JOBCTL, *AUDIT, *SPLCTL, *SERVICE, *IOSYSCFG
QSYSOPR	System operator profile	• USRCLS: *SYSOPR • SPCAUT: *ALLOBJ [1], *SAVSYS, *JOBCTL • INLMNU: SYSTEM • LIBRARY: *LIBL • MSGQ: QSYSOPR • DLVRY: *BREAK • SEV: 40 • ACGCDE: *BLANK
QTCM	Triggered cache manager profile	• STATUS: *DISABLED
QTCP	Transmission control protocol (TCP) profile	• USRCLS: *SYSOPR • SPCAUT: *JOBCTL • CCSID: *HEX • SRTSEQ: *HEX
QTFTP	Trivial File Transfer Protocol	

Table 134. IBM-Supplied User Profiles (continued)

Profile Name	Descriptive Name	Parameters Different from Default Values
QTMPLPD	Transmission control protocol/Internet protocol (TCP/IP) printing support profile	• PTYLMT: 3 • AUT: *USE
QTMPLPD	Remote LPR user profile	• JOBD: QGPL/QDFTJOBD • PWDEXPITV: *NOMAX • MSGQ: QTCP/QTMPLPD
QTMTWSG	HTML Workstation Gateway Profile user profile	• MSGQ: QUSRSYS/QTMTWSG • TEXT: HTML Workstation Gateway Profile
QTMHHTTP	HTML Workstation Gateway Profile user profile	• MSGQ: QUSRSYS/QTMHHTTP • TEXT: HTTP Server Profile
QTMHHTP1	HTML Workstation Gateway Profile user profile	• MSGQ: QUSRSYS/QTMHHTTP • TEXT: HTTP Server CGI Profile
QTSTRQS	Test request profile	
QUMB	Ultimedia System Facilities user profile	
QUMVUSER	Ultimedia Business Conferencing user profile	
QUSER	Workstation user profile	• PTYLMT: 3
QX400 user profile	OSI Messages Services File Services User Profile	• CURLIB: *QX400 • USRCLS: *SYSOPR • MSGQ: QX400/QX400 • DLVRY: *HOLD • OUTQ: *DEV • PRTDEV: *SYSVAL • ATNPGM: *NONE • CCSID: *HEX • TEXT: Internal OSI Messages Services User Profile
QYPSJSVR	Management Central Java Server profile	
QYPUOWN	Internal APU user profile	• TEXT: Internal APU — User profile

[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 135 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 116.

In Table 135, 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, "Authority Required for Objects Used by Commands" on page 309

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, "Authority Required for Objects Used by Commands" on page 309 for the object authorities required for commands.

Table 135. Authorities of IBM-Supplied User Profiles to Restricted Commands

Command Name	QSECOFR	QPGMR	QSYSOPR	QSRV	QSRVBAS
ADDCMDCRQA		S	S	S	S
ADDCRSDMNK	R				
ADDDSTQ		S	S		
ADDDSTRTE		S	S		
ADDDSTSYSN		S	S		
ADDEXITPGM	R				
ADDMFS	R				
ADDNETJOBE	R				
ADDOBJCRQA		S	S	S	S
ADDOPTCTG	R				
ADDOPTSVR	R				
ADDPEXDFN		S		S	
ADDPEXFTR		S		S	
ADDPRDCRQA		S	S	S	S
ADDPTFCRQA		S	S	S	S
ADDRPYLE		S			
ADDRSCCRQA		S	S	S	S

Table 135. Authorities of IBM-Supplied User Profiles to Restricted Commands (continued)

Command Name	QSECOFR	QPGMR	QSYSOPR	QSRV	QSRVBAS
ANSQST	R				
ANZACCGRP	R				
ANZBESTMDL	R				
ANZDBF	R				
ANZDBFKEY	R				
ANZDFTPWD	R				
ANZPFRDTA	R				
ANZPGM	R				
ANZPRB		S	S	S	S
ANZPRFACT	R				
ANZS34OCL	R				
ANZS36OCL	R				
APYJRNCHG		S		S	
APYPTF				S	
APYRMTPTF		S	S	S	S
CFGDSTSRV		S	S		
CFGRPDS		S	S		
CFGSYSSEC	R				
CHGACTSCDE	R				
CHGCMDCRQA		S	S	S	S
CHGCRSDMNK	R				
CHGDSTPWD[1]	R				
CHGDSTQ		S	S		
CHGDSTRTE		S	S		
CHGEXPSCDE	R				
CHGFCNARA	R				
CHGGPHFMT	R				
CHGGPHPKG	R				
CHGJOBTRC	R				
CHGJOBTYP	R				
CHGJRN		S	S	S	
CHGLICINF	R				
CHGMGDSYSA		S	S	S	S
CHGMGRSRVA		S	S	S	S
CHGMSTK	R				
CHGNETA	R				
CHGNETJOBE	R				
CHGNFSEXP	R				
CHGNWSA	R				

Table 135. Authorities of IBM-Supplied User Profiles to Restricted Commands (continued)

Command Name	QSECOFR	QPGMR	QSYSOPR	QSRV	QSRVBAS
CHGOBJCRQA		S	S	S	S
CHGOPTA	R				
CHGPEXDFN		S		S	
CHGPRB		S	S	S	S
CHGPRDCRQA		S	S	S	S
CHGPTFCRQA		S	S	S	S
CHGPTR				S	
CHGQSTDB	R				
CHGRCYAP		S	S		
CHGRPYLE		S			
CHGRSCCRQA		S	S	S	S
CHGSYSLIBL	R				
CHGSYSVAL		S	S	S	
CHGS34LIBM	R				
CHKCMNTRC				S	
CHKPRDOPT		S	S	S	S
CPHDTA	R				
CPYFCNARA	R				
CPYGPHFMT	R				
CPYGPHPKG	R				
CPYPFRDTA	R				
CPYPTF		S	S	S	S
CPYPTFGRP		S	S	S	S
CRTAUTHLR	R				
CRTBESTMDL	R				
CRTCLS	R				
CRTFCNARA	R				
CRTGPHFMT	R				
CRTGPHPKG	R				
CRTHSTDTA	R				
CRTJOBD	R				
CRTPFRDTA	R				
CRTLASREP		S			
CRTPEXDT		S		S	
CRTQSTDB	R				
CRTQSTLOD	R				
CRTSBSD		S	S		
CRTUDFS	R				
CRTUDFS	R				

Table 135. Authorities of IBM-Supplied User Profiles to Restricted Commands (continued)

Command Name	QSECOFR	QPGMR	QSYSOPR	QSRV	QSRVBAS
CRTVLDL	R				
CVTBASSTR	R				
CVTBASUNF	R				
CVTBGUDTA	R				
CVTPFRDTA	R				
CVTPFRTHD	R				
CVTS36CFG	R				
CVTS36FCT	R				
CVTS36JOB	R				
CVTS36QRY	R				
CVTS38JOB	R				
CVTTCPCL		S	S	S	S
DLTAPARDTA		S	S	S	S
DLTBESTMDL	R				
DLTCMNTRC				S	
DLTFCNARA	R				
DLTGPHFMT	R				
DLTGPHPKG	R				
DLTHSTDTA	R				
DLTLICPGM	R				
DLTPEXDTA		S		S	
DLTPFRDTA	R				
DLTPRB		S	S	S	S
DLTPTF		S	S	S	S
DLTQST	R				
DLTQSTDB	R				
DLTRMTPTF		S	S	S	S
DLTSMGOBJ		S	S	S	S
DLTUDFS	R				
DLTVLDL	R				
DMPDLO		S	S	S	S
DMPJOB		S	S	S	S
DMPJOBINT		S	S	S	S
DMPOBJ				S	S
DMPSYSOBJ		S	S	S	S
DMPTRC	R	S		S	
DSPACCGRP	R				
DSPAUDJRNE	R				
DSPDSTLOG	R				

Table 135. Authorities of IBM-Supplied User Profiles to Restricted Commands (continued)

Command Name	QSECOFR	QPGMR	QSYSOPR	QSRV	QSRVBAS
DSPHSTGPH	R				
DSPMFSINF	R				
DSPMGDSYSA		S	S	S	S
DSPPFRDTA	R				
DSPPFRGPH	R				
DSPPTF		S	S	S	S
DSPSRVSTS		S	S	S	S
DSPUDFS	R				
EDTCPCST			S		
EDTQST	R				
EDTRBDAP			S		
EDTRCYAP		S	S		
ENCCPHK	R				
ENCFRMMSTK	R				
ENCTOMSTK	R				
ENDCHTSVR	R				
ENDCMNTRC	R			S	
ENDDBGSVR		S	S	S	S
ENDHOSTSVR		S	S	S	S
ENDIDXMON	R				
ENDIPSIFC		S	S	S	S
ENDJOBABN		S	S	S	
ENDJOBTRC	R				
ENDMGDSYS		S	S	S	S
ENDMGRSRV		S	S	S	S
ENDMSF			S	S	S
ENDNFSSVR	R		S	S	S
ENDPEX		S		S	
ENDPFRTRC	R			S	
ENDSRVJOB		S	S	S	S
ENDSYSMGR		S	S	S	S
ENDTCP		S	S	S	S
ENDTCPCNN		S	S	S	S
ENDTCPIFC		S	S	S	S
ENDTCPSVR		S	S	S	S
GENCPHK	R				
GENCRSDMNK	R				
GENMAC	R				
GENPIN	R				

Table 135. Authorities of IBM-Supplied User Profiles to Restricted Commands (continued)

Command Name	QSECOFR	QPGMR	QSYSOPR	QSRV	QSRVBAS
GENS36RPT	R				
GENS38RPT	R				
GRTACCAUT	R				
HLDCMNDEV		S	S	S	S
HLDDSTQ		S	S		
INSPTF[3]				S	
INSRMTPRD		S	S	S	S
INZDSTQ		S	S		
INZSYS	R				
LODPTF				S	
LODQSTDB	R				
MGRS36	R				
MGRS36APF	R				
MGRS36CBL	R				
MGRS36DFU	R				
MGRS36DSPF	R				
MGRS36ITM	R				
MGRS36LIB	R				
MGRS36MNU	R				
MGRS36MSGF	R				
MGRS36QRY	R				
MGRS36RPG	R				
MGRS36SEC	R				
MGRS38OBJ	R				
MIGRATE	R				
PKGPRDDST		S	S	S	S
PRTACTRPT	R				
PRTADPOBJ	R				
PRTCMNSEC	R				
PRTCMNTRC				S	
PRTCPTRPT	R				
PRTJOBRPT	R				
PRTJOBTRC	R				
PRTLCKRPT	R				
PRTPOLRPT	R				
PRTRSCRPT	R				
PRTSYSRPT	R				
PRTTNSRPT	R				
PRTTRCRPT	R				

Table 135. Authorities of IBM-Supplied User Profiles to Restricted Commands (continued)

Command Name	QSECOFR	QPGMR	QSYSOPR	QSRV	QSRVBAS
PRTDSKINF	R				
PRTERRLOG		S	S	S	S
PRTINTDTA		S	S	S	S
PRTJOBDAUT	R				
PRTPRFINT	R				
PRTPVTAUT	R				
PRTQAUT	R				
PRTSBSDAUT	R				
PRTSYSSECA	R				
PRTTRGPGM	R				
PRTUSRPRF	R				
PRTUSROBJ	R				
PWRDWNSYS	R		S		
RCLOPT	R				
RCLSPLSTG	R				
RCLSTG		S	S	S	S
RCLTMPSTG		S	S	S	S
RESMGRNAM	R	S	S	S	S
RLSCMNDEV		S	S	S	S
RLSDSTQ		S	S		
RLSIFSLCK	R				
RLSRMTPHS		S	S		
RMVACC	R				
RMVCRSDMNK	R				
RMVDSTQ		S	S		
RMVDSTRTE		S	S		
RMVDSTSYSN		S	S		
RMVEXITPGM	R				
RMVJRNCHG		S		S	
RMVLANADP	R				
RMVMFS	R				
RMVNETJOBE	R				
RMVOPTCTG	R				
RMVOPTSVR	R				
RMVPEXDFN		S		S	
RMVPEXFTR		S		S	
RMVPTF				S	
RMVRMTPTF		S	S	S	S
RMVRPYLE		S			
RSTAUT	R				

Table 135. Authorities of IBM-Supplied User Profiles to Restricted Commands (continued)

Command Name	QSECOFR	QPGMR	QSYSOPR	QSRV	QSRVBAS
RST[4]					
RSTCFG	R				
RSTDLO	R				
RSTLIB	R				
RSTLICPGM	R				
RSTOBJ[4]					
RSTS36F	R				
RSTS36FLR	R				
RSTS36LIBM	R				
RSTS38AUT	R				
RSTUSFCNR[5]					
RSTUSRPRF	R				
RTVDSKINF	R				
RTVPRD		S	S	S	S
RTVPTF		S	S	S	S
RTVSMGOBJ		S	S	S	S
RUNLPDA		S	S	S	S
RUNSMGCMD		S	S	S	S
RUNSMGOBJ		S	S	S	S
RVKPUBAUT	R				
SAVAPARDTA		S	S	S	S
SAVLICPGM	R				
SBMFNCJOB	R				
SBMNWSCMD	R				
SETMSTK	R				
SNDDSTQ		S	S		
SNDPRD		S	S	S	S
SNDPTF		S	S	S	S
SNDPTFORD				S	S
SNDSMGOBJ		S	S	S	S
SNDSRVRQS				S	S
STRBEST	R				
STRCHTSVR	R				
STRCMNTRC				S	
STRDBG		S		S	S
STRDBGSVR		S	S	S	S
STRHOSTSVR		S	S	S	S
STRIDXMON	R				
STRIPSIFC		S	S	S	S
STRJOBTRC	R				

Table 135. Authorities of IBM-Supplied User Profiles to Restricted Commands (continued)

Command Name	QSECOFR	QPGMR	QSYSOPR	QSRV	QSRVBAS
STRMGDSYS		S	S	S	S
STRMGRSRV		S	S	S	S
STRMSF[2]			S	S	S
STRNFSSVR	R				
STRPEX		S		S	
STRPFRG	R				
STRPFRT	R				
STRPFRTRC	R			S	
STRRGZIDX	R				
STRSRVJOB		S	S	S	S
STRSST				S	
STRSYSMGR		S	S	S	S
STRS36MGR	R				
STRS38MGR	R				
STRTCP		S	S	S	S
STRTCPIFC		S	S	S	S
STRTCPSVR		S	S	S	S
STRUPDIDX	R				
TRCCPIC	R				
TRCICF	R				
TRCINT		S		S	
TRCJOB		S	S	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		
WRKFCNARA	R				
WRKJRN		S	S	S	
WRKLICINF	R				
WRKORDINF			S	S	
WRKPEXDFN		S		S	
WRKPEXFTR		S		S	
WRKPGMTBL	R				

Table 135. Authorities of IBM-Supplied User Profiles to Restricted Commands (continued)

Command Name	QSECOFR	QPGMR	QSYSOPR	QSRV	QSRVBAS
WRKPRB		S	S	S	S
WRKPTFGRP		S	S	S	S
WRKSRVPVD				S	S
WRKSYSACT	R				
WRKTXTIDX	R				
WRKUSRTBL	R				

[1] The CHGDSTPWD command is shipped with public authority *USE, but you must be signed on as QSECOFR to use this command.

[2] The QMSF user profile is also authorized to this command.

[3] QSRV can only run this command if an IPL is not being done.

[4] In addition to QSYS, user profile QRDARS400 has authority.

[5] In addition to QSYS, user profile QUMB has authority.

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 in "User Profile Commands" on page 436, 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" on page 311 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 136 describes the authorities that are specified in the *Authority Needed* column. The description includes examples of how the authority is used. In most cases, accessing an object requires a combination of object and data authorities.

Authority Needed for Library: This column shows what authority is needed for the library containing the object. For most operations, *EXECUTE authority is needed to locate the object in the library. Adding an object to a library usually requires *READ and *ADD authority. Table 136 describes the authorities that are specified in the *Authority Needed* column.

Table 136. Description of Authority Types

Authority	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. All functions defined for *OBJALTER and *OBJREF.
*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.
*OBJALTER	Object Alter	Add, clear, initialize and reorganize members of the database files. Alter and add attributes of database files: add and remove triggers. Change the attributes of SQL packages. Move a library or folder to a different ASP.

Table 136. Description of Authority Types (continued)

Authority	Name	Functions Allowed
*OBJREF	Object Reference	Specify a database file as the parent in a referential constraint. For example, you want to define a rule that a customer record must exist in the CUSMAS file before an order for the customer can be added to the CUSORD file. You need *OBJREF authority to the CUSMAS file to define this rule.
*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.
*ADD	Add	Add entries to an object, such as adding messages to a message 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.
*EXECUTE	Execute	Run a program, service program, or SQL package. Locate an object in a library or a directory.

[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 127 for more information.

In addition to these values, the *Authority Needed* columns of the table may show system-defined subsets of these authorities. Table 137 shows the subsets of object authorities and data authorities.

Table 137. System-Defined Authority

Authority	*ALL	*CHANGE	*USE	*EXCLUDE
Object Authorities				
*OBJOPR	X	X	X	
*OBJMGT	X			
*OBJEXIST	X			
*OBJALTER	X			
*OBJREF	X			
Data Authorities				
*READ	X	X	X	
*ADD	X	X		
*UPD	X	X		
*DLT	X	X		
*EXECUTE	X	X	X	

Table 138 on page 311 shows additional authority subsets that are supported by the CHGAUT and WRKAUT commands.

Table 138. System-Defined Authority

Authority	*RWX	*RW	*RX	*R	*WX	*W	*X
Object Authorities							
*OBJOPR	X	X	X	X	X	X	X
*OBJMGT							
*OBJEXIST							
*OBJALTER							
*OBJREF							
Data Authorities							
*READ	X	X	X	X			
*ADD	X	X			X	X	
*UPD	X	X			X	X	
*DLT	X	X			X	X	
*EXECUTE	X		X		X		X

For more information on these authorities and their descriptions, see "Defining How Information Can Be Accessed" on page 120.

Assumptions

1. To use any command, *USE authority is required to the command. This authority is not specifically listed in the tables.
2. 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 for Object Authorities on Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
Change (CHG) with F4 (Prompt)[7]	Current values	The current values are displayed if the user has authority to those values.	*EXECUTE
Command accessing object in directory	Directories in path prefix for QLANSrv file system	*R	
	Directories in path prefix for all other file systems	*X	
	Directory when pattern is specified (* or ?) for QLANSrv file system	None	
	Directory when pattern is specified (* or ?) for all other file system	*R	

General Rules for Object Authorities on Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
Copy (CPY) where to-file is a database file	Object to be copied	*OBJOPR, *READ	*EXECUTE
	CRTPF command, if CRTFILE (*YES) is specified	*OBJOPR	*EXECUTE
	To-file, if CRTFILE (*YES) is specified[1]		*ADD, *EXECUTE
	To-file, if it exists and new member is added	*OBJOPR, *OBJMGT, *ADD, *DLT	*ADD, *EXECUTE
	To-file, if file and member exist and *ADD option is specified	*OBJOPR, *ADD	*EXECUTE
	To-file, if file and member exist and *REPLACE option is specified	*OBJOPR, *OBJMGT, *ADD, *DLT	*EXECUTE
	To-file, if it exists, a new member is added, and *UPDADD option is specified.[8]	*OBJOPR, *OBJMGT, *ADD, *UPD	*EXECUTE
	To-file, if file and member exist and *UPDADD option is specified.[8]	*OBJOPR, *ADD, *UPD	*EXECUTE
Create (CRT)	Object to be created[2]		*READ, *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 [6, 9]	Object to be created (and replaced)[2]	*OBJMGT, *OBJEXIST, *READ[5]	*READ, *ADD
	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	*EXECUTE
	Output file, if file does not exist[3]		*ADD, *EXECUTE
	Output file, if file exists and new member is added, or if *REPLACE option specified and member did not previously exist	*OBJOPR, *OBJMGT, *ADD, *DLT	*ADD, *EXECUTE
	Output file, if file and member exist and *ADD option is specified	*OBJOPR, *ADD	*EXECUTE
	Output file, if file and member exist and *REPLACE option is specified	*OBJOPR, *OBJMGT, *ADD, *DLT	*EXECUTE
	Format file (QAxxxxx file in QSYS), if output file does not exist	*OBJOPR, *READ	
Display (DSP) using *PRINT or Work (WRK) using *PRINT	Object to be displayed	*USE	*EXECUTE
	Output queue[4]	*READ	*EXECUTE
	Printer file (QPxxxxx in QSYS)	*USE	*EXECUTE
Save (SAV) or other operation using device description	Device description	*USE	*EXECUTE
	Device file associated with device description, such as QSYSTAP for the TAP01 device description	*USE	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
[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, primary group 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, *OBJOPR authority is also required.			
[6] 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.			
[7] Authority to the corresponding (DSP) command is also required.			
[8] The *UPDADD option in only available on the MBROPT parameter of the CPYF command.			
[9] This does not apply to the REPLACE parameter on the CRTJVAPGM command.			

Commands Common for Most 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	For Library
ALCOBJ [1,2,11]	Object	*OBJOPR	*EXECUTE
ANZUSROBJ [20]			
CHGOBJAUD [18]	ASP Device (if specified)	*USE	
CHGOBJD [3]	Object, if it is a file	*OBJOPR, *OBJMGT	*EXECUTE
	Object, if it is not a file	*OBJMGT	*EXECUTE
CHGOBJOWN [3,4]	Object	*OBJEXIST	*EXECUTE
	Object (if file, library, subsystem description)	*OBJOPR, *OBJEXIST	*EXECUTE
	Object (if authorization list)	Ownership or *ALLOBJ	*EXECUTE
	Old user profile	*DLT	*EXECUTE
	New user profile	*ADD	*EXECUTE
	ASP Device (if specified)	*USE	

Commands Common for Most Objects

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGOBJPGP [3]	Object	*OBJEXIST	*EXECUTE
	Object (if file, library, subsystem description)	*OBJOPR, *OBJEXIST	*EXECUTE
	Object (if authorization list)	Ownership and *OBJEXIST, or *ALLOBJ	*EXECUTE
	Old user profile	*DLT	
	New user profile	*ADD	
	ASP Device (if specified)	*USE	
CHKOBJ [3]	Object	Authority specified by AUT parameter [14]	*EXECUTE
CPROBJ	Object	*OBJMGT	*EXECUTE
CHKOBJITG [11](Q)			
CRTDUPOBJ [3,9,11,21]	New object		*USE, *ADD
	Object being copied, if it is an authorization list	*AUTLMGT	*USE, *ADD
	Object being copied, all other types	*OBJMGT, *USE	*USE
	CRTSAVF command (if the object is a save file)	*OBJOPR	
DCPOBJ	Object	*USE	*EXECUTE
DLCOBJ [1,11]	Object	*OBJOPR	*EXECUTE
DMPOBJ (Q) [3]	Object	*OBJOPR, *READ	*EXECUTE
DMPSYSOBJ (Q)	Object	*OBJOPR, *READ	*EXECUTE
DSPOBJAUT [3]	Object (to see all authority information)	*OBJMGT or *ALLOBJ special authority or ownership	*EXECUTE
	Output file	See General Rules on page 311	See General Rules on page 311
	ASP Device (if specified)	*USE	
DSPOBJD [2]	Output file	See General Rules on page 281	See General Rules on page 281
EDTOBJAUT [3,5,6,15]	Object	*OBJMGT	*EXECUTE
	Object (if file)	*OBJOPR, *OBJMGT	*EXECUTE
	Authorization list, if used to secure object	Not *EXCLUDE	
	ASP Device (if specified)	*USE	
GRTOBJAUT [3,5,6,15]	Object	*OBJMGT	*EXECUTE
	Object (if file)	*OBJOPR, *OBJMGT	*EXECUTE
	Authorization list, if used to secure object	Not *EXCLUDE	
	ASP Device (if specified)	*USE	
	Reference ASP Device (if specified)	*EXECUTE	

Command	Referenced Object	Authority Needed	
		For Object	For Library
MOVOBJ [3,7,12]	Object	*OBJMGT	
	Object (if *File)	*OBJOPR, *OBJMGT	
	From-library		*CHANGE
	To-library		*READ, *ADD
PRTADPOBJ [140](Q)			
PRTPUBAUT [20]			
PRTUSROBJ [20]			
PRTPVTAUT [20]			
RCLSTG (Q)			
RCLTMPSTG (Q)	Object	*OBJMGT	*EXECUTE
RNMOBJ [3,11]	Object	*OBJMGT	*UPD, *EXECUTE
	Object, if authorization list	*AUTLMGT	*EXECUTE
	Object (if *FILE)	*OBJOPR, *OBJMGT	*UPD, *EXECUTE
RSTOBJ [3,13] (Q)	Object, if it already exists in the library	*OBJEXIST [8]	*EXECUTE, *ADD
	Media definition	*USE	*EXECUTE
	Message queues being restored to library where they already exist	*OBJOPR, *OBJEXIST [8]	*EXECUTE, *ADD
	User profile owning objects being created	*ADD [8]	
	Program that adopts authority	Owner or *SECADM and *ALLOBJ special authority	*EXECUTE
	To-library	*EXECUTE, *ADD [8]	
	Library for saved object if VOL(*SAVVOL) is specified	*USE [8]	
	Save file	*USE	*EXECUTE
RSTOBJ [3,13] (Q) (continued)	Tape unit, diskette unit or optical unit	*USE	*EXECUTE
	Tape (QSYSTAP) file or diskette (QSYSDKT) file	*USE [8]	*EXECUTE
	Optical File (OPTFILE)[22]	*R	N/A
	Parent Directory of optical file (OPTFILE)[22]	*X	N/A
	Path prefix of OPTFILE[22]	*X	N/A
	Optical volume[24]	*USE	N/A
	QSYS/QPSRLDSP print file, if OUTPUT(*PRINT) specified	*USE	*EXECUTE
	Output file, if specified	See General Rules on page 311	See General Rules on page 311
	QSYS/QASRRSTO field reference file for output file, if an output file is specified and does not exist	*USE	*EXECUTE
RVKPUBAUT [20]	Tape (QSYSTAP) file or diskette (QSYSDKT) file	*USE [8]	*EXECUTE
RTVOBJD [2]	Optical File (OPTFILE)[22]	*R	N/A
	Parent Directory of optical file (OPTFILE)[22]	*X	N/A

Commands Common for Most Objects

Command	Referenced Object	Authority Needed	
		For Object	For Library
RVKOBJAUT [3,5,15]	Path prefix of OPTFILE[22]	*X	N/A
	Optical volume[24]	*USE	N/A
	QSYS/QPSRLDSP print file, if OUTPUT(*PRINT) specified	*USE	*EXECUTE
	ASP Device (if specified)	*USE	
SAVCHGOBJ [3]	Output file, if specified	See General Rules on page 311	See General Rules on page 311
	QSYS/QASRRSTO field reference file for output file, if an output file is specified and does not exist	*USE	*EXECUTE
	Tape unit, diskette unit, optical unit	*USE	*EXECUTE
	Save file, if empty	*USE, *ADD	*EXECUTE
	Save file, if records exist in it	*OBJMGT, *USE, *ADD	*EXECUTE
	Save active message queue	*OBJOPR, *ADD	*EXECUTE
SAVCHGOBJ [3] (continued)	Optical File (OPTFILE)[22]	*RW	N/A
	Parent Directory of optical file (OPTFILE)[22]	*WX	N/A
	Path prefix of optical file (OPTFILE)[22]	*X	N/A
	Root Directory (/) of optical volume[22, 23]	*RWX	N/A
	Optical volume[24]	*CHANGE	
	Output file, if specified	See General Rules on page 311	See General Rules on page 311
	QSYS/QASAVOBJ field reference file for output file, if an output file is specified and does not exist	*USE [8]	*EXECUTE
	QSYS/QPSAVOBJ print file	*USE [8]	*EXECUTE
SAVOBJ [3]	Object	*OBJEXIST [8]	*EXECUTE
	Media definition	*USE	*EXECUTE
	Tape unit, diskette unit, optical unit	*USE	*EXECUTE
	Save file, if empty	*USE, *ADD	*EXECUTE
	Save file, if records exist in it	*OBJMGT, *USE, *ADD	*EXECUTE
	Save active message queue	*OBJOPR, *ADD	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
SAVOBJ [3] (continued)	Optical File (OPTFILE) [22]	*RW	N/A
	Parent Directory of optical file (OPTFILE)[22]	*WX	N/A
	Path prefix of OPTFILE[22]	*X	N/A
	Root directory (/) of optical volume [22, 23]	*RWX	N/A
	Optical volume[24]	*CHANGE	
	Output file, if specified	See General Rules on page 311	See General Rules on page 311
	QSYS/QASAVOBJ field reference file for output file, if an output file is specified and does not exist	*USE [8]	*EXECUTE
	QSYS/QPSAVOBJ print file	*USE [8]	*EXECUTE
SAVSTG [10]			
SAVSYS [10]	Tape unit, optical unit	*USE	*EXECUTE
	Root directory (/) of optical volume[22]	*RWX	N/A
	Optical volume[24]	*CHANGE	N/A
SAVRSTCHG	On the source system, same authority as required by SAVCHGOBJ command.		
	On the target system, same authority as required by RSTOBJ command.		
SAVRSTLIB	On the source system, same authority as required by SAVLIB command.		
	On the target system, same authority as required by RSTLIB command.		
SAVRSTOBJ	On the source system, same authority as required by SAVOBJ command.		
	On the target system, same authority as required by RSTOBJ command.		
SETOBJACC	Object	*OBJOPR	*EXECUTE
WRKOBJ [19]	Object	Any authority	*USE
WRKOBJLCK			
WRKOBJOWN [17]	User profile	*READ	*EXECUTE
WRKOBJPGP [17]	User profile	*READ	*EXECUTE

1 See the OBJTYPE keyword of the ALCOBJ command for the list of object types that can be allocated and deallocated.

2 Some authority to the object (other than *EXCLUDE) is required.

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, service program, or SQL package that adopts authority.

5 You must be the owner or have *OBJMGT authority and the authorities being granted or revoked.

Commands Common for Most Objects

Command	Referenced Object	Authority Needed	
		For Object	For Library
[6] You must be the owner or have *ALLOBJ special authority to grant *OBJMGT or *AUTLMGT authority. [7] This command cannot be used for user profiles, controller descriptions, device descriptions, line descriptions, documents, document libraries, and folders. [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. • If an error occurs while copying authorities to the new object, the newly created object is deleted. • *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 *OBJEXIST authority for FILEB, you must have *OBJEXIST authority to FILEB. [15] To secure an object with an authorization list or remove the authorization list from the object, you must (one of the following): • Own the object. • Have *ALL authority to the object. • Have *ALLOBJ special authority.			
[16] If either the original file or the renamed file has an associated authority holder, *ALL authority to the authority holder is required. [17] This command does not support the QOPT file system. [18] You must have *AUDIT special authority. [19] To use an individual operation, you must have the authority required by the individual operation. [20] You must have *ALLOBJ special authority.			
[21] All authorities on the from-object are duplicated to the new object. The primary group of the new object is determined by the group authority type (GRPAUTTYP) field in the user profile that is running the command. If the from-object has a primary group, the new object may not have the same primary group, but the authority that the primary group has on the from-object will be duplicated to the new object. [22] This authority check is only made when the Optical media format is Universal Disk Format. [23] This authority check is only made if you are clearing the optical volume [24] Optical volumes are not actual system objects. The link between the optical volume and the authorization list used to secure the volume is maintained by the optical support function.			

Authorities Needed

Access Path Recovery Commands

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.

These commands do not require object authorities.			
CHGRCYAP [1, 3] (Q)	DSPRCYAP [1]	EDTRBDAP [2] (Q)	EDTRCYAP [1, 3] (Q)

[1] You must have *JOBCTL special authority to use this command.

[2] You must have *ALLOBJ special authority to use this command.

[3] You must have *USE authority to access IASP device description.

Advanced Function Printing™ Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
ADDFNTTBLE	Font table	*CHANGE	*EXECUTE
CHGCDEFNT	Font resource	*CHANGE	*EXECUTE
CHGFNTTBLE	Font table	*CHANGE	*EXECUTE
CRTFNTRSC	Source file	*USE	*EXECUTE
	Font resource: REPLACE(*NO)		*READ, *ADD
	Font resource: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
CRTFNTTBL	Font table		*READ, *ADD
CRTFORMDF	Source file	*USE	*EXECUTE
	Form definition: REPLACE(*NO)		*READ, *ADD
	Form definition: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
CRTOVL	Source file	*USE	*EXECUTE
	Overlay: REPLACE(*NO)		*READ, *ADD
	Overlay: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
CRTPAGDFN	Source file	*USE	*EXECUTE
	Page definition: REPLACE(*NO)		*READ, *ADD
	Page definition: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
CRTPAGSEG	Source file	*USE	*EXECUTE
	Page segment: REPLACE(*NO)		*READ, *ADD
	Page segment: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
DLTFNTRSC	Font resource	*OBJEXIST	*EXECUTE
DLTFNTTBL	Font table	*CHANGE	*EXECUTE
DLTFORMDF	Form definition	*OBJEXIST	*EXECUTE

Printing Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
DLTOVL	Overlay	*OBJEXIST	*EXECUTE
DLTPAGDFN	Page definition	*OBJEXIST	*EXECUTE
DLTPAGSEG	Page segment	*OBJEXIST	*EXECUTE
DSPCDEFNT	Font resource	*USE	*EXECUTE
DSPFNTRSCA	Font resource	*USE	*EXECUTE
DSPFNTTBL	Font table	*USE	*EXECUTE
RMVFNTTBLE	Font table	*CHANGE	*EXECUTE
WRKFNTRSC [1]	Font resource	*USE	*USE
WRKFORMDF [1]	Form definition	*USE	*USE
WRKOVL [1]	Overlay	*USE	*USE
WRKPAGDFN [1]	Page definition	Any authority	*USE
WRKPAGSEG [1]	Page segment	*USE	Any authority
[1] To use individual operations, you must have the authority required by the individual operation.			

AF_INET Sockets Over SNA Commands

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.

These commands do not require any authority to objects:			
ADDIPSIFC[1]	CHGIPSIFC[1]	CVTIPSLOC	RMVIPSLOC[1]
ADDIPSRTE[1]	CHGIPSLOC[1]	ENDIPSIFC (Q)	RMVIPSRTE[1]
ADDIPSLOC[1]	CHGIPSTOS[1]	PRTIPSCFG	STRIPSIFC (Q)
CFGIPS	CVTIPSIFC	RMVIPSIFC[1]	
[1] You must have *IOSYSCFG special authority to use this command.			

Alerts

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 to others.

Command	Referenced Object	Authority Needed	
		For Object	For Library
ADDALRD	Alert table	*USE, *ADD	*EXECUTE
CHGALRD	Alert table	*USE, *UPD	*EXECUTE
CHGALRTBL (Q)	Alert table	*CHANGE	*EXECUTE
CRTALRTBL (Q)	Alert table		*READ, *ADD
DLTALR	Physical file QAALERT	*USE, *DLT	*EXECUTE
DLTALRTBL (Q)	Alert table	*OBJEXIST	*EXECUTE
RMVALRD	Alert table	*USE, *DLT	*EXECUTE
WRKALR [1]	Physical file QAALERT	*USE	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
WRKALRD [1]	Alert table	*USE	*EXECUTE
WRKALRTBL [1]	Alert table	*READ	*USE

[1] To use individual operations, you must have the authority required by the individual operation.

Application Development Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
EXPPART	Object to be created by export, if it does not already exist		*USE, *ADD
	Object to be created by export, if it is being replaced	*OBJOPR, *OBJMGT, *OBJEXIST	*USE, *ADD, *DLT
	Source file to which part is being exported, if the member does not exist	*OBJOPR, *OBJMGT, *ADD	*USE, *ADD
	Source file to which part is being exported, if the member is being replaced	*OBJOPR, *OBJMGT, *OBJEXIST, *ADD, *DLT	*USE, *ADD
	User profile to which ownership of exported part is assigned	*ADD	
IMPPART	Any importable objects	*OBJMGT, *USE	*USE
	Source file, when importing a source member	*USE	*USE
FNDSTRPDM	Source part	*READ	*EXECUTE
MRGFORMD	Form description	*READ	*EXECUTE
STRAPF [1]	Source file	*OBJMGT, *CHANGE	*READ, *ADD
	Commands CRTPF, CRTLF, ADDPFM, ADDLFM, and RMVM	*USE	*EXECUTE
STRBGU [1]	Chart	*OBJMGT, *CHANGE	*EXECUTE
STRDFU [1]	Program (if create program option)		*READ, *ADD
	Program (if change or delete program option)	*OBJEXIST	*EXECUTE
	Program (if change or display data option)	*USE	*EXECUTE
	Database file (if change data option)	*OBJOPR, *ADD, *UPD, *DLT	*EXECUTE
	Database file (if display data option)	*USE	*EXECUTE
	Display file (if display or change data option)	*USE	*EXECUTE
	Display file (if change program option)	*USE	*EXECUTE
	Display file (if delete program option)	*OBJEXIST	*EXECUTE
STRPDM [1]			

Application Development Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
STRRLU	Source file	*READ, *ADD, *UPD, *DLT	*EXECUTE
	Edit, add, or change a member	*OBJOPR, *OBJMGT	*READ, *ADD
	Browse a member	*OBJOPR	*EXECUTE
	Print a prototype report	*OBJOPR	*EXECUTE
	Remove a member	*OBJOPR, *OBJEXIST	*EXECUTE
	Change type or text of member	*OBJOPR	*EXECUTE
STRSDA	Source file	*READ, *ADD, *UPD, *DLT	*EXECUTE
	Update and add new member	*CHANGE, *OBJMGT	*READ, *ADD
	Delete member	*ALL	*EXECUTE
STRSEU [1]	Source file	*USE	*EXECUTE
	Edit or change a member	*CHANGE, *OBJMGT	*EXECUTE
	Add a member	*USE, *OBJMGT	*READ, *ADD
	Browse a member	*USE	*EXECUTE
	Print a member	*USE	*EXECUTE
	Remove a member	*USE, *OBJEXIST	*EXECUTE
	Change type or text of a member	*USE, *OBJMGT	*EXECUTE
WRKGRPPDM [1,4]	Group [2]	*READ	*EXECUTE
WRKLIBPDM [1]			
WRKMBRPDM [1]	Source file	*USE	*EXECUTE
WRKOBJPDM [1]	File	*READ	*EXECUTE
WRKPARTPDM [1,4]	Part (object or source member)	*READ	*EXECUTE
WRKPRJPDM [1,4]	Project [3]	*READ	*EXECUTE

[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 *WebSphere Development Studio: Application Development Manager User's Guide* book.

Authority Holder Commands

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	For Library
CRTAUTHLR (Q)	Associated object if it exists	*ALL	*EXECUTE
DLTAUTHLR	Authority holder	*ALL	*EXECUTE
DSPAUTHLR	Output file	See General Rules on page 311	See General Rules on page 311

Authorization List Commands

Command	Referenced Object	Authority Needed	
		For Object	For QSYS Library
ADDAUTLE [1]	Authorization list	*AUTLMGT or ownership	*EXECUTE
CHGAUTLE [1]	Authorization list	*AUTLMGT or ownership	*EXECUTE
CRTAUTL			
DLTAUTL	Authorization list	Owner or *ALLOBJ	*EXECUTE
DSPAUTL	Authorization list		*EXECUTE
	Output file	See General Rules on page 311	See General Rules on page 311
DSPAUTLDLO	Authorization list	*USE	*EXECUTE
DSPAUTLOBJ	Authorization list	*READ	*EXECUTE
	Output file	See General Rules on page 311	See General Rules on page 311
EDTAUTL [1]	Authorization list	*AUTLMGT or ownership	*EXECUTE
RMVAUTLE [1]	Authorization list	*AUTLMGT or ownership	*EXECUTE
RTVAUTLE [2]	Authorization list	*AUTLMGT or ownership	*EXECUTE
WRKAUTL [3,4,5]	Authorization list		

1 You must be the owner or have authorization list management authority and have the authorities being given or taken away.

2 If do not have *OBJMGT or *AUTLMGT, you can retrieve *PUBLIC authority and your own authority. You must have *READ authority to your own profile to retrieve your own authority.

3 To use an individual operation, you must have the authority required by the operation

4 You must not be excluded (*EXCLUDE) from the authorization list.

5 Some authority to the authorization list is required.

Binding Directory Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
ADDBNDDIRE	Binding directory	*OBJOPR, *ADD	*USE
CRTBNDDIR	Binding directory		*READ, *ADD
DLTBNDDIR	Binding directory	*OBJEXIST	*EXECUTE
DSPBNDDIR	Binding directory	*READ, *OBJOPR	*USE
RMVBNDDIRE	Binding directory	*OBJOPR, *DLT	*READ, *OBJOPR
WRKBNDDIR [1]	Binding directory	Any authority	*USE
WRKBNDDIRE [1]	Binding directory	*READ, *OBJOPR	*USE

1 To use individual operations, you must have the authority required by the operation.

Change Request Description Commands

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	For Library
ADDCMDCRQA (Q)	Change request description	*CHANGE	*EXECUTE
ADDOBJCRQA (Q)	Change request description	*CHANGE	*EXECUTE
ADDPRDCRQA (Q)	Change request description	*CHANGE	*EXECUTE
ADDPTFCRQA (Q)	Change request description	*CHANGE	*EXECUTE
ADDRSCCRQA (Q)	Change request description	*CHANGE	*EXECUTE
CHGCMDCRQA (Q)	Change request description	*CHANGE	*EXECUTE
CHGOBJCRQA (Q)	Change request description	*CHANGE	*EXECUTE
CHGPRDCRQA (Q)	Change request description	*CHANGE	*EXECUTE
CHGPTFCRQA (Q)	Change request description	*CHANGE	*EXECUTE
CHGCRQD	Change request description	*CHANGE	*EXECUTE
CHGRSCCRQA (Q)	Change request description	*CHANGE	*EXECUTE
CRTCRQD	Change request description		*READ, *ADD
DLTCRQD	Change request description	*OBJEXIST	*EXECUTE
RMVCRQDA	Change request description	*CHANGE	*EXECUTE
WRKCRQD [1]	Change request description		*EXECUTE

[1] To use an individual operation, you must have the authority required by the operation

Chart Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
DLTCHTFMT	Chart format	*OBJEXIST	*EXECUTE
DSPCHT	Chart format	*USE	*USE
	Database file	*USE	*USE
DSPGDF	Database file	*USE	*USE
STRBGU (Option 3) [2]	Chart format	*CHANGE, *OBJEXIST	*EXECUTE
WRKCHTFMT [1]	Chart format	Any authority	*USE

[1] To use an individual operation, you must have the authority required by the operation .

[2] Option 3 on the BGU menu (shown when STRGBU is run) is the Change chart format option.

Class Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGCLS	Class	*OBJMGT, *OBJOPR	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTCLS	Class		*READ, *ADD
DLTCLS	Class	*OBJEXIST	*EXECUTE
DSPCLS	Class	*OBJOPR	*EXECUTE
WRKCLS [1]	Class	*OBJOPR	*USE

[1] To use an individual operation, you must have the authority required by the operation .

Class-of-Service Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGCOSD [3]	Class-of-service description	*CHANGE, OBJMGT	*EXECUTE
CRTCOSD [3]	Class-of-service description		
DLTCOSD	Class-of-service description	*OBJEXIST	*EXECUTE
DSPCOSD	Class-of-service description	*USE	*EXECUTE
WRKCOSD [1,2]	Class-of-service description	*OBJOPR	*EXECUTE

[1] To use individual operations, you must have the authority required by the individual operation.

[2] Some authority to the object is required.

[3] To use this command, you must have *IOSYSCFG special authority.

Command (*CMD) Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGCMD	Command	*OBJMGT	*EXECUTE
CHGCMDDFT	Command	*OBJMGT, *USE	*EXECUTE
CRTCMD	Source file	*USE	*EXECUTE
	Command: REPLACE(*NO)		*READ, *ADD
	Command: REPLACE(*YES)	See General Rules on page 311	See General Rules on page 311
DLTCMD	Command	*OBJEXIST	*EXECUTE
DSPCMD	Command	*USE	*EXECUTE
SBMRMTCMD	Command	*OBJOPR	*EXECUTE
	DDM file	*USE	*EXECUTE
SLTCMD [1]	Command	Any authority	*USE
WRKCMD [2]	Command	Any authority	*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.

Commitment Control Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
COMMIT			
ENDCMTCTL	Message queue, as specified on NFYOBJ keyword for the associated STRCMTCTL command.	*OBJOPR, *ADD	*EXECUTE
ROLLBACK			
STRCMTCTL	Message queue, when specified on NFYOBJ keyword	*OBJOPR, *ADD	*EXECUTE
	Data area, as specified on NFYOBJ keyword for the associateed STRCMTCTL command.	*CHANGE	*EXECUTE
	Files, as specified on NFYOBJ keyword for the associated STRCMTCTL command.	*OBJOPR *READ	*EXECUTE
WRKCMTDFN [1]			

[1] Any user can run this command for commitment definitions that belong to a job that is running under the user profile of the user. A user who has job control (*JOBCTL) special authority can run this command for any commitment definition.

Communications Side Information Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGCSI	Communications side information object	*USE, *OBJMGT	*EXECUTE
	Device description [1]	*CHANGE	
CRTCSI	Communications side information object		*READ, *ADD
	Device description [1]	*CHANGE	
DLTCSI	Communications side information object	*OBJEXIST	*EXECUTE
DSPCSI	Communications side information object	*READ	*EXECUTE
WRKCSI	Communications side information objects	*USE	*EXECUTE

[1] Authority is verified when the communications side information object is used.

Configuration Commands

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	For Library
PRTDEVADR	Controller description (CTL)	*USE	*EXECUTE
	Device description	*USE	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
RSTCFG (Q) [5]	Every object being restored over by a saved version	*OBJEXIST [1]	*EXECUTE
	To-library		*ADD, *EXECUTE [1]
	User profile owning objects being created	*ADD [1]	
	Tape unit	*USE	*EXECUTE
	Tape file (QSYSTAP)	*USE [1]	*EXECUTE
	Save file, if specified	*USE	*EXECUTE
	Print file (QPSRLDSP), if output(*print) is specified	*USE	*EXECUTE
	Output file, if specified	See General Rules on page 311	See General Rules on page 311
	QSYS/QASRRSTO field reference file, if output file is specified and it does not exist	*USE	*EXECUTE
RTVCFGSTS	Object	*OBJOPR	*EXECUTE
RTVCFGSRC	Object	*USE	*EXECUTE
	Source file	*OBJOPR, *OBJMGR, *ADD, *DLT	*EXECUTE
SAVCFG [2]	Save file, if empty	*USE, *ADD	*EXECUTE
	Save file, if records exist in it	*USE, *ADD, *OBJMGT	*EXECUTE
SAVRSTCFG	On the source system, same authority as required by SAVCFG command.		
	On the target system, same authority as required by RSTCFG command.		
VRYCFG [3,6]	Object	*USE, *OBJMGT	*EXECUTE
WRKCFGSTS [4]	Object	*OBJOPR	*EXECUTE

1 If you have *SAVSYS special authority, you do not need the authority specified.

2 You must have *SAVSYS special authority.

3 If a user has *JOBCTL special authority, authority to the device is not needed.

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).

6 You must have *IOSYSCFG special authority for media library when status is *ALLOCATE or *DEALLOCATE.

Configuration List Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
ADDCFGLE [2]	Configuration list	*CHANGE, *OBJMGT	*EXECUTE
CHGCFGL [2]	Configuration list	*CHANGE, *OBJMGT	*EXECUTE
CHGCFGLE [2]	Configuration list	*CHANGE, *OBJMGT	*EXECUTE
CPYCFGL [2]	Configuration list	*USE, *OBJMGT	*ADD

Configuration List Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTCFGL [2]	Configuration list		
DLTCFGL	Configuration list	*OBJEXIST	*EXECUTE
DSPCFGL [2]	Configuration list	*USE, *OBJMGT	*EXECUTE
RMVCFGLE [2]	Configuration list	*CHANGE, *OBJMGT	*EXECUTE
WRKCFGL [1, 2]	Configuration list	*OBJOPR	*EXECUTE

1 To use the individual operations, you must have the authority required by the individual operation.

2 To use this command, you must have *IOSYSCFG special authority.

Connection List Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
ADDCNNLE [2]	Connection list	*CHANGE, *OBJMGT	*EXECUTE
CHGCNNL [2]	Connection list	*CHANGE, *OBJMGT	*EXECUTE
CHGCNNLE [2]	Connection list	*CHANGE, *OBJMGT	*EXECUTE
CRTCNNL [2]			*EXECUTE
DLTCNNL	Connection list	*OBJEXIST	*EXECUTE
DSPCNNL	Connection list	*USE	*EXECUTE
RMVCNNLE [2]	Connection list	*CHANGE, *OBJMGT	*EXECUTE
RNMCNNLE [2]	Connection list	*CHANGE, *OBJMGT	*EXECUTE
WRKCNNL [1]	Connection list	*OBJOPR	*EXECUTE
WRKCNNLE [1]	Connection list	*USE	*EXECUTE

1 To use the individual operations, you must have the authority required by the individual operation.

2 To use this command, you must have *IOSYSCFG special authority.

Controller Description Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGCTLAPPC [2]	Controller description	*CHANGE, *OBJMGT	*EXECUTE
	Line description (SWTLINLST)	*USE	*EXECUTE
	Connection list (CNNLSTOUT)	*USE	*EXECUTE
CHGCTLASC [2]	Controller description	*CHANGE, *OBJMGT	*EXECUTE
	Line description (SWTLINLST)	*USE	*EXECUTE
CHGCTLBSC [2]	Controller description	*CHANGE, *OBJMGT	*EXECUTE
	Line description (SWTLINLST)	*USE	*EXECUTE
CHGCTLFNC [2]	Controller description	*CHANGE, *OBJMGT	*EXECUTE
	Line description (SWTLINLST)	*USE	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGCTLHOST [2]	Controller description	*CHANGE, *OBJMGT	*EXECUTE
	Line description (SWTLINLST)	*USE	*EXECUTE
	Connection list (CNNLSTOUT)	*USE	*EXECUTE
CHGCTLLWS [2]	Controller description	*CHANGE, *OBJMGT	*EXECUTE
	Program (INZPGM)	*USE	*EXECUTE
CHGCTLNET [2]	Controller description	*CHANGE, *OBJMGT	*EXECUTE
CHGCTLRTL [2]	Controller description	*CHANGE, *OBJMGT	*EXECUTE
	Line description (SWTLINLST)	*USE	*EXECUTE
CHGCTLRWS [2]	Controller description	*CHANGE, *OBJMGT	*EXECUTE
	Line description (SWTLINLST)	*USE	*EXECUTE
	Connection list (CNNLSTOUT)	*USE	*EXECUTE
CHGCTLTAP [2]	Controller description	*CHANGE, *OBJMGT	*EXECUTE
CHGCTLVWS [2]	Controller	*CHANGE, *OBJMGT	*EXECUTE
CRTCTLAPPC [2]	Line description (LINE or SWTLINLST)	*USE	*EXECUTE
	Device description (DEV)	*USE	*EXECUTE
	Connection list (CNNLSTOUT)	*USE	*EXECUTE
	Controller description		
CRTCTLASC [2]	Line description (LINE or SWTLINLST)	*USE	*EXECUTE
	Device description (DEV)	*USE	*EXECUTE
	Controller description		
CRTCTLBSC [2]	Line description (LINE or SWTLINLST)	*USE	*EXECUTE
	Device description (DEV)	*USE	*EXECUTE
	Controller description		
CRTCTLFNC [2]	Line description (LINE or SWTLINLST)	*USE	*EXECUTE
	Device description (DEV)	*USE	*EXECUTE
	Controller description		
CRTCTLHOST [2]	Line description (LINE or SWTLINLST)	*USE	*EXECUTE
	Device description (DEV)	*USE	*EXECUTE
	Connection list (CNNLSTOUT)	*USE	*EXECUTE
	Controller description		
CRTCTLLWS [2]	Device description (DEV)	*USE	*EXECUTE
	Controller description		
	Program (INZPGM)	*USE	*EXECUTE
CRTCTLNET [2]	Line description (LINE)	*USE	*EXECUTE
	Device description (DEV)	*USE	*EXECUTE
	Controller description		
CRTCTLRTL [2]	Line description (LINE or SWTLINLST)	*USE	*EXECUTE
	Device description (DEV)	*USE	*EXECUTE
	Controller description		

Controller Description Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTCTLRWS [2]	Line description (LINE or SWTLINLST)	*USE	*EXECUTE
	Device description (DEV)	*USE	*EXECUTE
	Connection list (CNNLSTOUT)	*USE	*EXECUTE
	Controller description		
CRTCTLTAP [2]	Device description (DEV)	*USE	*EXECUTE
	Controller description		
CRTCTLVWS [2]	Device description (DEV)	*USE	*EXECUTE
	Controller description		
DLTCTLD	Controller description	*OBJEXIST	*EXECUTE
DSPCTLD	Controller description	*USE	*EXECUTE
ENDCTLRCY	Controller description	*USE	*EXECUTE
PRTCMNSEC [2, 3]			
RSMCTLRCY	Controller description	*USE	*EXECUTE
WRKCTLD [1]	Controller description	*OBJOPR	*EXECUTE

[1] To use the individual operations, you must have the authority required by the individual operation.

[2] To use this command, you must have *IOSYSCFG special authority.

[3] To use this command, you must have *ALLOBJ special authority.

Cryptography Commands

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	For Library
ADDCRSDMNK (Q)	QUSRSYS/QACRKTBL *FILE	*OBJOPR, *ADD	*EXECUTE
	QHST message queue	*OBJOPR, *ADD	*EXECUTE
CHGCRSDMNK (Q)	QUSRSYS/QACRKTBL *FILE	*OBJOPR, *READ, *UPD	*EXECUTE
	QHST message queue	*OBJOPR, *ADD	*EXECUTE
CHGMSTK (Q)	QUSRSYS/QACRKTBL *FILE	*OBJOPR, *READ, *UPD	*EXECUTE
	QHST message queue	*OBJOPR, *ADD	*EXECUTE
CPHDTA (Q)			
ENCCPHK (Q)			
ENCFRMMSTK (Q)	QUSRSYS/QACRKTBL *FILE	*OBJOPR, *ADD	*EXECUTE
ENCTOMSTK (Q)	QUSRSYS/QACRKTBL *FILE	*OBJOPR, *READ	*EXECUTE
GENCPHK (Q)	QUSRSYS/QACRKTBL *FILE	*OBJOPR, *READ	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
GENCRSDMNK (Q)	QUSRSYS/QACRKTBL *FILE	*OBJOPR, *ADD	*EXECUTE
	QCRP/QPCRGENX *FILE	*OBJOPR, *READ	*EXECUTE
	QHST message queue	*OBJOPR, *ADD	*EXECUTE
GENMAC (Q)			
GENPIN (Q)	QUSRSYS/QACRKTBL *FILE	*OBJOPR, *READ	*EXECUTE
RMVCRSDMNK (Q)	QUSRSYS/QACRKTBL *FILE	*OBJOPR, *READ, *DLT	*EXECUTE
	QHST message queue	*OBJOPR, *ADD	*EXECUTE
SETMSTK (Q)	QUSRSYS/QACRKTBL *FILE	*OBJOPR, *READ, *UPD	*EXECUTE
	QHST message queue	*OBJOPR, *ADD	*EXECUTE
TRNPIN (Q)	QUSRSYS/QACRKTBL *FILE	*OBJOPR, *READ	*EXECUTE
VFYMSTK (Q)	QHST message queue	*OBJOPR, *ADD	*EXECUTE
VFYPIN (Q)	QUSRSYS/QACRKTBL *FILE	*OBJOPR, READ	*EXECUTE

Data Area Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGDTAARA [1]	Data area	*CHANGE	*EXECUTE
CRTDTAARA [1]	Data area		*READ, *ADD
	APPC device description[4]	*CHANGE	
DLTDTAARA	Data area	*OBJEXIST	*EXECUTE
DSPDTAARA	Data area	*OBJOPR	*EXECUTE
RTVDTAARA [2]	Data area	*OBJOPR	*EXECUTE
WRKDTAARA [3]	Data area	Any authority	*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 To use an individual operation, you must have the authority required by the operation.

4 Authority is verified when the data area is used.

Data Queue Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTDTAQ	Data queue		*READ, *ADD
	Target data queue for the QSNDDTAQ program	*OBJOPR, *ADD	*EXECUTE
	Source data queue for the QRCVDTAQ program	*OBJOPR, *READ	*EXECUTE
	APPC device description[2]	*CHANGE	
DLTDTAQ	Data queue	*OBJEXIST	*EXECUTE
WRKDTAQ [1]	Data queue	*READ	*USE

[1] To use individual operations, you must have the authority required by the individual operation.

[2] Authority is verified when the data area is used.

Device Description Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CFGDEVMLB [4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
CHGDEVAPPC [4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
	Mode description (MODE)	*USE	*EXECUTE
CHGDEVASC [4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
CHGDEVASP[4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
CHGDEVBSC [4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
CHGDEVDKT [4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
CHGDEVDSP [4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
	Printer (PRINTER)	*USE	*EXECUTE
CHGDEVFNC [4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
CHGDEVHOST [4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
CHGDEVINTR [4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
CHGDEVMLB [4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
CHGDEVNET [4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
CHGDEVOPT [4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
CHGDEVPRT [4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
CHGDEVRTL [4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
CHGDEVSNPT [4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
CHGDEVSNUF [4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
CHGDEVTAP [4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
CRTDEVAPPC [4]	Controller description (CTL)	*USE	*EXECUTE
	Device description		
	Mode description (MODE)	*USE	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTDEVASC [4]	Controller description (CTL)	*USE	*EXECUTE
	Device description		
CRTDEVASP [4]	Device description		*EXECUTE
CRTDEVBSC [4]	Controller description (CTL)	*USE	*EXECUTE
	Device description		
CRTDEVDKT [4]	Device description		
CRTDEVDSP [4]	Printer description (PRINTER)	*USE	*EXECUTE
	Controller description (CTL)	*USE	*EXECUTE
	Device description		
CRTDEVFNC [4]	Controller description (CTL)	*USE	*EXECUTE
	Device description		
CRTDEVHOST [4]	Controller description (CTL)	*USE	*EXECUTE
	Device description		
CRTDEVINTR [4]	Device description		
CRTDEVMLB [4]	Device description		*EXECUTE
CRTDEVNET [4]	Controller description (CTL)	*USE	*EXECUTE
	Device description		
CRTDEVOPT [4]	Device description		*EXECUTE
CRTDEVPRT [4]	Controller description (CTL)	*USE	*EXECUTE
	Device description		
CRTDEVRTL [4]	Controller description (CTL)	*USE	*EXECUTE
	Device description		
CRTDEVSNPT [4]	Controller description (CTL)	*USE	*EXECUTE
	Device description		
CRTDEVSNUF [4]	Controller description (CTL)	*USE	*EXECUTE
	Device description		
CRTDEVTAP [4]	Controller description (CTL)	*USE	*EXECUTE
	Device description		
DLTDEVD [1]	Device description	*OBJEXIST	*EXECUTE
DSPCNNSTS	Device description	*OBJOPR	*EXECUTE
DSPDEVD	Device description	*USE	*EXECUTE
ENDDEVRCY	Device description	*USE	*EXECUTE
HLDCMNDEV [2]	Device description	*OBJOPR	*EXECUTE
PRTCMNSEC [4, 5]			
RLSCMNDEV	Device description	*OBJOPR	*EXECUTE
RSMDEVRCY	Device description	*USE	*EXECUTE
WRKDEVD [3]	Device description	*OBJOPR	*EXECUTE

Device Description Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library

[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.

[4] You must have *IOSYSCFG special authority to run this command.

[5] You must have *ALLOBJ special authority to run this command.

Device Emulation Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
ADDEMLCFGE	Emulation configuration file	*CHANGE	*EXECUTE
CHGEMLCFGE	Emulation configuration file	*CHANGE	*EXECUTE
EJTEMLOUT	Emulation device description when specified	*OBJOPR	*EXECUTE
	Emulation device description when location specified	*OBJOPR	*EXECUTE
ENDPRTEML	Emulation device description when specified	*OBJOPR	*EXECUTE
	Emulation device description when location specified	*OBJOPR	*EXECUTE
EMLPRTKEY	Emulation device description when specified	*OBJOPR	*EXECUTE
	Emulation device description when location specified	*OBJOPR	*EXECUTE
EML3270	Emulation device description	*OBJOPR	*EXECUTE
	Emulation controller description	*OBJOPR	*EXECUTE
RMVEMLCFGE	Emulation configuration file	*CHANGE	*EXECUTE
STREML3270	Emulation configuration file	*OBJOPR	*EXECUTE
	Emulation device, emulation controller description, display station device, and display station controller description	*OBJOPR	*EXECUTE
	Printer device description, user exit program, and translation tables when specified	*OBJOPR	*EXECUTE
STRPRTEML	Emulation configuration file	*OBJOPR	*EXECUTE
	Emulation device description and emulation controller description	*OBJOPR	*EXECUTE
	Printer device description, print file, message queue, job description, job queue, and translation tables when specified	*OBJOPR	*EXECUTE
SNDEMLIGC	From-file	*OBJOPR	*EXECUTE
TRMPRTEML	Emulation device description	*OBJOPR	*EXECUTE

Directory and Directory Shadowing Commands

These commands do not require any object authorities:			
ADDDIRE [2]	CHGDIRSHD [1]	ENDDIRSHD [4]	STRDIRSHD [4]
ADDDIRSHD [1]	CPYFRMDIR [1]	RMVDIRE [1]	WRKDIRE [3,5]
CHGSYSDIRA [2]	CPYTODIR [1]	RMVDIRSHD [1]	WRKDIRLOC [1,5]
CHGDIRE [3]	DSPDIRE	RNMDIRE [2]	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.

Disk Commands

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.

These commands to not require authority to any objects:		
ENDDSKRGZ (Q) [1]	STRDSKRGZ (Q) [1]	WRKDSKSTS

[1] To use this command, you must have *ALLOBJ special authority.

Display Station Pass-Through Commands

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.

		Authority Needed	
Command	**Referenced Object**	**For Object**	**For Library**
ENDPASTHR			
STRPASTHR	APPC device on source system	*CHANGE	*EXECUTE
	APPC device on target system	*CHANGE	*EXECUTE
	Virtual controller on target system [1]	*USE	*EXECUTE
	Virtual device on target system [1,2]	*CHANGE	*EXECUTE
	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

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	For Library
ADDDSTQ (Q)			
ADDDSTRTE (Q)			
ADDDSTSYSN (Q)			
CFGDSTSRV (Q)			
CFGRPDS (Q)			
CHGDSTD [1]	Document [2]	*CHANGE	*EXECUTE
CHGDSTQ (Q)			
CHGDSTRTE (Q)			
DLTDST [1]			
DSPDSTLOG (Q)	Journal	*USE	*EXECUTE
	Journal receiver	*USE	*EXECUTE
DSPDSTSRV (Q)			
HLDDSTQ (Q)			
INZDSTQ (Q)			
QRYDST [1]	Requested file	*CHANGE	*EXECUTE
RCVDST [1]	Requested file	*CHANGE	*EXECUTE
	Folder	*CHANGE	*EXECUTE
RLSDSTQ (Q)			
RMVDSTQ (Q)			
RMVDSTRTE (Q)			
RMVDSTSYSN (Q)			
SNDDST [1]	Requested file or document	*USE	*EXECUTE
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 Commands

These commands do not require any object authorities:			
ADDDSTLE [1]	CRTDSTL	DSPDSTL	RNMDSTL [1]
CHGDSTL[1]	DLTDSTL [1]	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 Object Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
ADDDLOAUT	Document library object	*ALL or owner	*EXECUTE
CHGDLOAUD [1]			
CHGDLOAUT	Document library object	*ALL or owner	*EXECUTE
CHGDLOOWN	Document library object	Owner or *ALLOBJ special authority	*EXECUTE
	Old user profile	*DLT	*EXECUTE
	New user profile	*ADD	*EXECUTE
CHGDLOPGP	Document library object	Owner or *ALLOBJ special authority	*EXECUTE
	Old primary group profile	*DLT	*EXECUTE
	New primary group profile	*ADD	*EXECUTE
CHGDOCD [2]	Document description	*CHANGE	*EXECUTE
CHKDLO [2]	Document library object	As required by the AUT keyword	*EXECUTE
CHKDOC	Document	*CHANGE	*EXECUTE
	Spelling aid dictionary	*CHANGE	*EXECUTE
CPYDOC	From-document	*USE	*EXECUTE
	To-document, if replacing existing document	*CHANGE	*EXECUTE
	To-folder if to-document is new	*CHANGE	*EXECUTE
CRTDOC	In-folder	*CHANGE	*EXECUTE
CRTFLR	In-folder	*CHANGE	*EXECUTE
DLTDLO [3]	Document library object	*ALL	*EXECUTE
DLTDOCL	Document list	*ALL [4]	*EXECUTE
DMPDLO [15]			
DSPAUTLDLO	Authorization list	*USE	*EXECUTE
	Document library object	*USE	*EXECUTE
DSPDLOAUD	Output file, if specified	See General Rules on page 311	See General Rules on page 311
DSPDLOAUT	Document library object	*USE or owner	*EXECUTE
DSPDLONAM	Document library object	*USE	*EXECUTE
DSPDOC	Document	*USE	*EXECUTE
DSPFLR	Folder	*USE	*EXECUTE
EDTDLOAUT	Document library object	*ALL or owner	*EXECUTE
EDTDOC	Document	*CHANGE	*EXECUTE
FILDOC [2]	Requested file	*USE	*EXECUTE
	Folder	*CHANGE	*EXECUTE

Document Library Object Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
MOVDOC	From-folder, if source document is in a folder	*CHANGE	*EXECUTE
	From-document	*ALL	*EXECUTE
	To-folder	*CHANGE	*EXECUTE
MRGDOC [5]	Document	*USE	*EXECUTE
	From-folder	*USE	*EXECUTE
	To-document if document is replaced	See General Rules on page 311	See General Rules on page 311
	To-folder if to-document is new	See General Rules on page 311	See General Rules on page 311
PAGDOC	Document	*CHANGE	*EXECUTE
PRTDOC	Folder	*USE	*EXECUTE
	Document	*USE	*EXECUTE
	DLTPF, DLTF, and DLTOVR commands, if an *INDEX* instruction is specified	*USE	*EXECUTE
	CRTPF, OVRPRTF, DLTSPLF, and DLTOVR commands, if a *RUN* instruction is specified	*USE	*EXECUTE
	Save document, if SAVOUTPUT (*YES) is specified	*USE	*EXECUTE
	Save folder, if SAVOUTPUT (*YES) is specified	*USE	*EXECUTE
QRYDOCLIB [2,6]	Requested file	*USE	*EXECUTE
	Document list, if it exists	*CHANGE	*EXECUTE
RCLDLO	Document library object		
	Internal documents or all documents and folders[16]		
RGZDLO	Document library object	*CHANGE or owner	*EXECUTE
	DLO(*MAIL), DLO(*ANY), or DLO(*FLR) [16]		
RMVDLOAUT	Document library object	*ALL or owner	*EXECUTE
RNMDLO	Document library object	*ALL	*EXECUTE
	In-folder	*CHANGE	*EXECUTE
RPLDOC [2]	Requested file	*READ	*EXECUTE
	Document	*CHANGE	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
RSTDLO	Document library object, if replacing	*ALL[10]	*EXECUTE
	Parent folder, if new DLO	*CHANGE[10]	*EXECUTE
	Owning user profile, if new DLO	*ADD[10]	*EXECUTE
	Output file, if specified	See General Rules on page 311	See General Rules on page 311
	Save file	*USE	*EXECUTE
	Optical file (OPTFILE)[17]	*R	N/A
	Path prefix of optical file (OPTFILE)[17]	*X	N/A
	Optical volume[19]	*USE	N/A
	Tape, diskette, and optical unit	*USE	*EXECUTE
RSTS36FLR [11,12,14]	S/36 folder	*USE	*EXECUTE
	To-folder	*CHANGE	*EXECUTE
	Device file or device description	*USE	*EXECUTE
RTVDLONAM	Document library object	*USE	*EXECUTE
RTVDOC [2]	Document if checking out	*CHANGE	*EXECUTE
	Document if not checking out	*USE	*EXECUTE
	Requested file	*CHANGE	*EXECUTE
SAVDLO [7,13]	Document library object	*ALL[10]	*EXECUTE
	Tape unit, diskette unit, and optical unit	*USE	*EXECUTE
	Save file, if empty	*USE, *ADD	*EXECUTE
	Save file, if records exist in it	*USE, *ADD, *OBJMGT	*EXECUTE
	Output file, if specified	See General Rules on page 311	See General Rules on page 311
	Optical File (OPTFILE)[17]	*RW	N/A
	Parent directory of optical file (OPTFILE)[17]	*WX	N/A
	Path Prefix of optical file (OPTFILE)[17]	*X	N/A
	Root Directory (/) of volume[17, 18]	*RWX	N/A
	Optical Volume[19]	*CHANGE	N/A
SAVRSTDLO	On the source system, same authority as required by SAVDLO command.		
	On the target system, same authority as required by RSTDLO command.		
WRKDOC	Folder	*USE	
WRKFLR	Folder	*USE	

Document Library Object Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
[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.			
[17] This authority check is only made when the Optical Media Format is Universal Disk Format (UDF).			
[18] This authority check is only made when you are clearing the optical volume.			
[19] Optical volumes are not actual system objects. The link between the optical volume and the authorization list used to secure the volume is maintained by the optical support function.			

Double-Byte Character Set Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CPYIGCTBL	DBCS sort table (*IN)	*ALL	*EXECUTE
	DBCS sort table (*OUT)	*USE	*EXECUTE
CRTIGCDCT	DBCS conversion dictionary		*READ, *ADD
DLTIGCDCT	DBCS conversion dictionary	*OBJEXIST	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
DLTIGCSRT	DBCS sort table	*OBJEXIST	*EXECUTE
DLTIGCTBL	DBCS font table	*OBJEXIST	*EXECUTE
DSPIGCDCT	DBCS conversion dictionary	*USE	*EXECUTE
EDTIGCDCT	DBCS conversion dictionary	*USE, *UPD	*EXECUTE
	User dictionary	*ADD, *DLT	*EXECUTE
STRCGU	DBCS sort table	*CHANGE	*EXECUTE
	DBCS font table	*CHANGE	*EXECUTE
STRFMA	DBCS font table, if copy-to option specified	*OBJOPR, *READ *ADD, *UPD	*EXECUTE
	DBCS font table, if copy-from option specified	*OBJOPR, *READ	*EXECUTE
	Font management aid work file (QGPL/QAFSVDF)	*CHANGE	*EXECUTE

Edit Description Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTEDTD	Edit description		*EXECUTE, *ADD
DLTEDTD	Edit description	*OBJEXIST	*EXECUTE
DSPEDTD	Edit description	*OBJOPR	*EXECUTE
WRKEDTD [1]	Edit description	Any authority	*USE

1 To use an individual operation, you must have the authority required by the operation .

Environment Variable Commands

These commands do not require any object authorities.			
ADDENVVAR[1]	CHGENVVAR [1]	RMVENVVAR[1]	WRKENVVAR[1]

1 To update system-level environment variables, you need *JOBCTL special authority.

Extended Wireless LAN Configuration Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
ADDEWCBCDE	Source file	*USE	*EXECUTE
ADDEWCM	Source file	*USE	*EXECUTE
ADDEWCPTCE	Source file	*USE	*EXECUTE
ADDEWLM	Source file	*USE	*EXECUTE
CHGEWCBCDE	Source file	*USE	*EXECUTE
CHGEWCM	Source file	*USE	*EXECUTE
CHGEWCPTCE	Source file	*USE	*EXECUTE

Extended Wireless LAN Configuration Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGEWLM	Source file	*USE	*EXECUTE
DSPEWCBCDE	Source file	*USE	*EXECUTE
DSPEWCM	Source file	*USE	*EXECUTE
DSPEWCPTCE	Source file	*USE	*EXECUTE
DSPEWLM	Source file	*USE	*EXECUTE
RMVEWCBCDE	Source file	*USE	*EXECUTE
RMVEWCPTCE	Source file	*USE	*EXECUTE

File Commands

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	For Library
ADDICFDEVE	ICF file	*OBJOPR, *OBJMGT	*EXECUTE
ADDLFM	Logical file	*OBJOPR, *OBJMGT or *OBJALTER	*EXECUTE, *ADD
	File referenced in DTAMBRS parameter, when logical file is keyed	*OBJOPR, *OBJMGT or *OBJALTER	*EXECUTE
	File referenced in DTAMBRS parameter, when logical file is not keyed	*OBJOPR	*EXECUTE
ADDPFCST	Dependent file, if TYPE(*REFCST) is specified	*OBJMGT or *OBJALTER	*EXECUTE
	Parent file, if TYPE(*REFCST) is specified	*OBJMGT or *OBJREF	*EXECUTE
	File, if TYPE(*UNQCST) or TYPE(*PRIKEY) is specified	*OBJMGT	*EXECUTE
ADDPFM	Physical file	*OBJOPR, *OBJMGT or *OBJALTER	*EXECUTE, *ADD
ADDPFTRG	Physical file, to insert trigger	*OBJALTER, *OBJMGT, *READ, *OBJOPR	*EXECUTE
	Physical file, to delete trigger	*OBJALTER, *OBJMGT, *READ, *OBJOPR	*EXECUTE
	Physical file, to update trigger	*OBJALTER, *OBJMGT, *READ, *OBJOPR	*EXECUTE
	Trigger program	*EXECUTE	*EXECUTE
CHGDDMF	DDM file	*OBJOPR, *OBJMGT	*EXECUTE
	Device description [7]	*CHANGE	
CHGDKTF	Diskette file	*OBJOPR, *OBJMGT	*EXECUTE
	Device if device name specified in the command	*OBJOPR	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGDSPF	Display file	*OBJOPR, *OBJMGT	*EXECUTE
	Device if device name specified	*OBJOPR	*EXECUTE
CHGDTA	Data file	*OBJOPR, *ADD, *UPD, *DLT	*EXECUTE
	Program	*USE	*EXECUTE
	Display file	*USE	*EXECUTE
CHGICFDEVE	ICF file	*OBJOPR, *OBJMGT	*EXECUTE
CHGICFF	ICF file	*OBJOPR, *OBJMGT	*EXECUTE
CHGLF	Logical file	*OBJMGT or *OBJALTER	*EXECUTE
CHGLFM	Logical file	*OBJMGT or *OBJALTER	*EXECUTE
CHGPF	Physical file	*OBJMGT or *OBJALTER	*EXECUTE
CHGPFCST	Dependent file	*OBJMGT or *OBJALTER	*EXECUTE
CHGPFM	Physical file	*OBJMGT or *OBJALTER	*EXECUTE
CHGPFTRG	Physical file	*OBJMGT or *OBJALTER	*EXECUTE
CHGPRTF	Print file	*OBJOPR, *OBJMGT	*EXECUTE
	Device if device name specified	*OBJOPR	*EXECUTE
CHGSAVF	Save file	*OBJOPR, *OBJMGT	*EXECUTE
CHGSRCPF	Source physical file	*OBJMGT or *OBJALTER	*EXECUTE
CHGTAPF	Tape file	*OBJOPR, *OBJMGT	*EXECUTE
	Device if device name specified	*OBJOPR	*EXECUTE
CLRPFM	Physical file	*OBJOPR, *OBJMGT or *OBJALTER, *DLT	*EXECUTE
CLRSAVF	Save file	*OBJOPR, *OBJMGT	*EXECUTE
CPYF	From-file	*OBJOPR, *READ	*EXECUTE
	To-file (device file)	*OBJOPR, *READ	*EXECUTE
	To-file (physical file)	See General Rules on page 311	See General Rules on page 311
	Based-on file if from-file is logical file	*READ	*EXECUTE
CPYFRMDKT	From-file	*OBJOPR, *READ	*EXECUTE
	To-file (device file)	*OBJOPR, *READ	*EXECUTE
	To-file (physical file)	See General Rules on page 311	See General Rules on page 311

File Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CPYFRMIMPF	From-file	*OBJOPR, *READ	*USE
	To-file (device file)	*OBJOPR, *READ	*USE
	To-file (physical file)	See General Rules on page 311	See General Rules on page 311
	Based-on file if from-file is logical file	*READ	*USE
CPYFRMQRYF [1]	From-file	*OBJOPR, *READ	*EXECUTE
	To-file (device file)	*OBJOPR, *READ	*EXECUTE
	To-file (physical file)	See General Rules on page 311	See General Rules on page 311
CPYFRMSTMF	Stream file	*R	
	Directories in stream file path name prefix	*X	
	Target database file, if MBROPT(*ADD) specified	*X, *ADD	*X
	Target database file, if MBROPT(*REPLACE) specified	*X, *ADD, *DLT, *OBJMGT	*X
	Target database file, if new member created	*X, *OBJMGT, or *OBJALTER	*X, *ADD
	Conversion table *TBL used to translate data	*OBJOPR	*X
CPYFRMTAP	From-file	*OBJOPR, *READ	*EXECUTE
	To-file (device file)	*OBJOPR, *READ	*EXECUTE
	To-file (physical file)	See General Rules on page 311	See General Rules on page 311
CPYSRCF	From-file	*OBJOPR, *READ	*EXECUTE
	To-file (device file)	*OBJOPR, *READ	*EXECUTE
	To-file (physical file)	See General Rules on page 311	See General Rules on page 311
CPYTODKT	To-file and from-file	*OBJOPR, *READ	*EXECUTE
	Device if device name specified on the command	*OBJOPR, *READ	*EXECUTE
	Based-on physical file if from-file is logical file	*READ	*EXECUTE
CPYTOIMPF	From-file	*OBJOPR, *READ	*USE
	To-file (device file)	*OBJOPR, *READ	*USE
	To-file (physical file)	See General Rules on page 311	See General Rules on page 311
	Based-on file if from-file is logical file	*READ	*USE

Command	Referenced Object	Authority Needed	
		For Object	For Library
CPYTOSTMF	Database file	*RX	*X
	Stream file, if it already exists	*W	
	Stream file parent directory, if the stream file does not exist	*WX,	
	Document parent folder, if the document parent folder does not exist	*RWX	
	Stream file path name prefix	*X	
	Conversion table *TBL used to translate data	*OBJOPR	*X
CPYTOTAP	To-file and from file	*OBJOPR, *READ	*EXECUTE
	Device if device name is specified	*OBJOPR, *READ	*EXECUTE
	Based-on physical file if from-file is logical file	*READ	*EXECUTE
CRTDDMF	DDM file: REPLACE(*NO)		*READ, *ADD
	DDM file: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	Device description [7]	*CHANGE	
CRTDKTF	Device if device name is specified	*OBJOPR	*EXECUTE
	Diskette file: REPLACE(*NO)		*READ, *ADD, *EXECUTE
	Diskette file: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD, *EXECUTE
CRTDSPF	Source file	*USE	*EXECUTE
	Device if device name is specified	*OBJOPR	*EXECUTE
	File specified in REF and REFFLD keywords	*OBJOPR	*EXECUTE
	Display file: REPLACE(*NO)		*READ, *ADD, *EXECUTE
	Display file: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD, *EXECUTE
CRTICFF	Source file	*USE	*EXECUTE
	File specified in REF and REFFLD keywords	*OBJOPR	*EXECUTE
	ICF file: REPLACE(*NO)		*READ, *ADD
	ICF file: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD

File Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTLF	Source file	*USE	*EXECUTE
	File specified on PFILE or JFILE keyword, when logical file is keyed	*OBJOPR, *OBJMGT or *OBJALTER	*EXECUTE
	File specified on PFILE or JFILE keyword, when logical file is not keyed	*OBJOPR	*EXECUTE
	Files specified on FORMAT and REFACCPTH keywords	*OBJOPR	*EXECUTE
	Tables specified in the ALTSEQ keyword	*OBJOPR	*EXECUTE
	Logical file		*EXECUTE, *ADD
	File referenced in DTAMBRS parameter, when logical file is keyed	*OBJOPR, *OBJMGT or *OBJALTER	*EXECUTE
	File referenced in DTAMBRS parameter, when logical file is not keyed	*OBJOPR	*EXECUTE
CRTPF	Source file	*USE	*EXECUTE
	Files specified in FORMAT and REFFLD keywords and tables specified in the ALTSEQ keyword	*OBJOPR	*EXECUTE
	Physical file		*EXECUTE, *ADD
CRTPRTF	Source file	*USE	*EXECUTE
	Device if device name is specified	*OBJOPR	*EXECUTE
	Files specified in the REF and REFFLD keywords	*OBJOPR	*EXECUTE
	Print file: Replace(*NO)		*READ, *ADD, *EXECUTE
	Print file: Replace(*YES)	See General Rules on page 311	*READ, *ADD, *EXECUTE
CRTSAVF	Save file		*READ, *ADD, *EXECUTE
CRTSRCPF	Source physical file		*READ, *ADD, *EXECUTE
CRTS36DSPF	To-file source file when TOMBR is not *NONE	*ALL	*CHANGE
	Source file QS36SRC	*USE	*EXECUTE
	Display file: REPLACE(*NO)		*READ, *ADD
	Display file: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	Create Display File (CRTDSPF) command	*OBJOPR	*EXECUTE
CRTTAPF	Tape file: REPLACE(*NO)		*READ, *ADD
	Tape file: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	Device if device name is specified	*OBJOPR	*EXECUTE
DLTF	File	*OBJOPR, *OBJEXIST	*EXECUTE
DSPCPCST	Database file that has constraint pending	*OBJOPR, *READ	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
DSPDBR	Database file	*OBJOPR	*EXECUTE
	Output file, if specified	See General Rules on page 311	See General Rules on page 311
DSPDDMF	DDM file	*OBJOPR	
DSPDTA	Data file	*USE	*EXECUTE
	Program	*USE	*EXECUTE
	Display file	*USE	*EXECUTE
DSPFD [2]	File	*OBJOPR	*EXECUTE
	Output file	See General Rules on page 311	See General Rules on page 311
	File is a physical file and TYPE(*ALL, *MBR, OR *MBRLST) is specified	A data authority other than *EXECUTE	*EXECUTE
DSPFFD	File	*OBJOPR	*EXECUTE
	Output file	See General Rules on page 311	See General Rules on page 311
DSPPFM	Physical file	*USE	*EXECUTE
DSPSAVF	Save file	*USE	*EXECUTE
EDTCPCST	Data area, as specified on NFYOBJ keyword for the associated STRCMTCTL command.	*CHANGE	*EXECUTE
	Files, as specified on NFYOBJ keyword for the associated STRCMTCTL command.	*OBJOPR, *ADD	*EXECUTE
GENCAT	Database file	*OBJOPR and a data authority other than *EXECUTE	*EXECUTE
INZPFM	Physical file, when RECORD(*DFT) is specified	*OBJOPR, *OBJMGT or *OBJALTER, *ADD	*EXECUTE
	Physical file, when RECORD(*DLT) is specified	*OBJOPR, *OBJMGT or *OBJALTER, *ADD, *DLT	*EXECUTE
MRGSRC	Target file	*CHANGE, *OBJMGT	*CHANGE
	Maintenance file	*USE	*EXECUTE
	Root file	*USE	*EXECUTE
OPNDBF	Database file	*OBJOPR and a data authority other than *EXECUTE	*EXECUTE
OPNQRYF	Database file	*OBJOPR and a data authority other than *EXECUTE	*EXECUTE
PRTTRGPGM [6]			
RGZPFM	File containing member	*OBJOPR, *OBJMGT or *OBJALTER, *READ, *ADD, *UPD, *DLT, *EXECUTE	*EXECUTE
RMVICFDEVE	ICF file	*OBJOPR, *OBJMGT	*EXECUTE
RMVM	File containing member	*OBJEXIST, *OBJOPR	*EXECUTE

File Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
RMVPFCST	File	*OBJMGT or *OBJALTER	*EXECUTE
RMVPFTRG	Physical file	*OBJALTER, *OBJMGT	*EXECUTE
RNMM	File containing member	*OBJOPR, *OBJMGT	*EXECUTE, *UPD
RSTS36F [4] (Q)	To-file	*ALL	See General Rules on page 311
	From-file	*USE	*EXECUTE
	Based on physical file, if file being restored is a logical (alternative) file	*CHANGE	*EXECUTE
	Device description for diskette or tape	*USE	*EXECUTE
RTVMBRD	File	*USE	*EXECUTE
SAVSAVFDTA	Tape, diskette, or optical device description	*USE	*EXECUTE
	Save file	*USE	*EXECUTE
	Optical Save/Restore File[8] (if previously exists)	*RW	N/A
	Parent Directory of OPTFILE[8]	*WX	N/A
	Path Prefix of OPTFILE[8]	*X	N/A
	Root Directory (/) of Optical Volume [8,9]	*RWX	N/A
	Optical Volume[10]	*CHANGE	N/A
SAVS36F	From-file	*USE	*EXECUTE
	To-file, when it is a physical file	*ALL	See General Rules on page 311
	Device file or device description	*USE	*EXECUTE
SAVS36LIBM	To-file, when it is a physical file	*ALL	See General Rules on page 311
	From-file	*USE	*EXECUTE
	Device file or device description	*USE	*EXECUTE
STRAPF [3]	Source file	*OBJMGT, *CHANGE	*READ, *ADD
	Commands CRTPF, CRTLF, ADDPFM, ADDLFM, and RMVM	*USE	*EXECUTE
	Data area, when specified on NFYOBJ keyword	*CHANGE	*EXECUTE
	Files, when specified on NFYOBJ keyword	*OBJOPR, *ADD	*EXECUTE
	Journal, when specified on DFTJRN keyword	*OBJOPR, *ADD	*EXECUTE
STRDFU [3]	Program (if create program option)		*READ, *ADD
	Program (if change or delete program option)	*OBJEXIST	*READ, *ADD
	File (if change or display data option)	*OBJOPR, *ADD, *UPD, *DLT	*EXECUTE
	File (if display data option)	*READ	*EXECUTE
UPDDTA	File	*CHANGE	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
WRKCMTDFN [1]			
WRKDDMF [3]	DDM file	*OBJOPR, *OBJMGT, *OBJEXIST	*READ, *ADD
WRKF [3,5]	Files	*OBJOPR	*USE
WRKPFCST[3]			*EXECUTE

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 311.)

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 Some authority to the object is required.

6 You must have *ALLOBJ special authority.

7 Authority is verified when the DDM file is used.

8 This authority check is only made when the Optical media format is Universal Disk Format (UDF).

9 This authority check is only made if you are clearing the optical volume.

10 Optical volumes are not actual system objects. The link between the optical volume and the authorization list used to secure the volume is maintained by the optical support function.

Filter Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
ADDALRACNE	Filter	*USE, *ADD	*EXECUTE
ADDALRSLTE	Filter	*USE, *ADD	*EXECUTE
ADDPRBACNE	Filter	*USE, *ADD	*EXECUTE
ADDPRBSLTE	Filter	*USE, *ADD	*EXECUTE
CHGALRACNE	Filter	*USE, *UPD	*EXECUTE
CHGALRSLTE	Filter	*USE, *UPD	*EXECUTE
CHGFTR	Filter	*OBJMGT	*EXECUTE
CHGPRBACNE	Filter	*USE, *UPD	*EXECUTE
CHGPRBSLTE	Filter	*USE, *UPD	*EXECUTE
CRTFTR	Filter		*READ, *ADD
DLTFTR	Filter	*OBJEXIST	*EXECUTE
RMVFTRACNE	Filter	*USE, *DLT	*EXECUTE
RMVFTRSLTE	Filter	*USE, *DLT	*EXECUTE
WRKFTR [1]	Filter	Any authority	*EXECUTE

Filter Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
WRKFTRACNE [1]	Filter	*USE	*EXECUTE
WRKFTRSLTE [1]	Filter	*USE	*EXECUTE

[1] To use an individual operation, you must have the authority required by the operation.

Finance Commands

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	For Library
SBMFNCJOB (Q)	Job description and message queue [1]	*OBJOPR	*EXECUTE
SNDFNCIMG (Q)	Job description and message queue [1]	*OBJOPR	*EXECUTE
WRKDEVTBL (Q)	Device description [1]	At least one data authority	*EXECUTE
WRKPGMTBL (Q)			
WRKUSRTBL (Q)			

[1] The QFNC user profile must have this authority.

OS/400 Graphical Operations

Command	Referenced Object	Authority Needed	
		For Object	For Library
EDTWSOAUT	Workstation object [1]	*OBJMGT [2,3,4]	*EXECUTE
GRTWSOAUT	Workstation object [1]	*OBJMGT [2,3,4]	*EXECUTE
RVKWSOAUT	Workstation object [1]	*OBJMGT [2,3,4]	*EXECUTE
SETCSTDTA	Copy-from user profile	*CHANGE	*EXECUTE
	Copy-to user profile	*CHANGE	*EXECUTE

[1] The workstation object is an internal object that is created when you install the OS/400 Graphical Operations feature. It is shipped with public authority of *USE.

[2] You must be the owner or have *OBJMGT authority and the authorities being granted or revoked.

[3] You must be the owner of have *ALLOBJ authority to grant *OBJMGT or *AUTLMGT authority.

[4] To secure the workstation object with an authorization list or remove the authorization list, you must have one of the following:

Own the workstation object.

Have *ALL authority to the workstation object.

Have *ALLOBJ special authority.

Graphics Symbol Set Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTGSS	Source file	*USE	*EXECUTE
	Graphics symbol set		*READ, *ADD
DLTGSS	Graphics symbol set	*OBJEXIST	*EXECUTE
WRKGSS [1]	Graphics symbol set	*OBJOPR	*USE
[1] Ownership or some authority to the object is required.			

Host Server Commands

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.

These commands do not require object authorities.	
ENDHOSTSVR (Q)	STRHOSTSVR (Q)

Integrated File System Commands

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	Object Type	File System	Authority Needed for Object[1]
ADDLNK	Object	*STMF	QOpenSys, "root" , UDFS	*OBJEXIST
		*FILE		*OBJMGT
	Parent of new link	*DIR[18]	QOpenSys, "root", UDFS	*WX
	Path prefix	See General Rules on page 311		

Integrated File System Commands

Command	Referenced Object	Object Type	File System	Authority Needed for Object[1]
CHGATR	Object when setting an attribute other than *USECOUNT, *ALWCKPWRT, *DISKSTGOPT, or *MAINSTGOPT	Any	All except QSYS.LIB	*W
	Object when setting *USECOUNT, DISKSTGOPT, or *MAINSTGOPT	Any	All except QSYS.LIB	*OBJMGT
		*FILE	QSYS.LIB	*R, *W, or *X plus *OBJMGT
		*MBR	QSYS.LIB	*X, *OBJMGT (authority inherited from parent *FILE)
		other	QSYS.LIB	*OBJMGT
	Object when setting *ALWCKPWRT	Any	All	*OBJMGT
	Directory that contains objects when SUBTREE(*ALL) is specified	Any directory	All	*RX
CHGAUD [4]				
CHGAUT	Object	All	QOpenSys, 'root', UDFS	Ownership[15]
			QSYS.LIB, QOPT[11]	Ownership or *ALLOBJ
			QDLS	Ownership, *ALL, or *ALLOBJ
				*OBJMGT
	Optical volume	*DDIR	QOPT[8]	*CHANGE
CHGCURDIR	Object specified by DIR parameter	Any directory		*R
	Optical volume	*DDIR	QOPT[8]	*X
CHGOWN	Object	All	QSYS.LIB	*OBJEXIST
		*FILE, *LIB[18], *SBSD	QSYS.LIB	*OBJEXIST, *OBJOPR
		All	QOpenSys, 'root' UDFS	Ownership and *OBJEXIST[15]
		All	QDLS	Ownership or *ALLOBJ
			QOPT[11]	Ownership or *ALLOBJ
CHGOWN (continued)	User profile of old owner—all except QOPT	*USRPRF	All	*DLT
	User profile of new owner—all except QOPT	*USRPRF	All	*ADD
	Optical volume	*DDIR	QOPT[8]	*CHANGE

Command	Referenced Object	Object Type	File System	Authority Needed for Object[1]
CHGPGP	Object	All	QSYS.LIB	*OBJEXIST
		*FILE, *LIB[18], *SBSD	QSYS.LIB	*OBJEXIST, *OBJOPR
		All	QOpenSys, 'root' UDFS	Ownership[5, 15]
		All	QDLS	Ownership or *ALLOBJ
			QOPT[11]	Ownership or *ALLOBJ
CHGPGP (continued)	User profile of old primary group—all except QOPT	*USRPRF	All	*DLT
	User profile of new primary group—all except QOPT	*USRPRF	All	*ADD
	Optical volume	*DDIR	QOPT[8]	*CHANGE
CHKIN	Object, if the user who checked it out.	*STMF	QOpenSys, 'root' UDFS	*W
		*DOC	QDLS	*W
	Object, if not the user who checked it out.	*STMF	QOpenSys, 'root' UDFS	*All or *ALLOBJ or Ownership
		*DOC	QDLS	*All or Ownership
CHKIN (continued)	Path, if not the user who checked out	*DIR[18]	QOpenSys, 'root' UDFS	*X
		*FLR	QDLS	None
	Path prefix	See General Rules on page 311		
CHKOUT	Object	*STMF	QOpenSys, 'root' UDFS	*W
		*DOC	QDLS	*W
	Path prefix	See General Rules on page 311		

Integrated File System Commands

Command	Referenced Object	Object Type	File System	Authority Needed for Object[1]
CPY	Object being copied, origin object	Any	QOpenSys, 'root' UDFS	*R, and *OBJMGT or ownership
		*DOC	QDLS	*RWX and *ALL or ownership
		*MBR	QSYS.LIB	None
		others	QSYS.LIB	*RX, *OBJMGT
		*DSTMF	QOPT[11]	*R
	Destination object when REPLACE(*YES) specified (if destination object already exists)	Any	All[10]	*W, *OBJEXIST, *OBJMGT
		*DSTMF	QOPT[11]	*W
	Directory being copied that contains objects when SUBTREE(*ALL) is specified, so that its contents are copied	*DIR[18]	QOpenSys, 'root', UDFS	*RX, *OBJMGT
CPY (continued)	Path (target), parent directory of destination object	*FILE	QSYS.LIB	*RX, *OBJMGT
		*LIB[18]	QSYS.LIB	*RX, *ADD
		*DIR[18]	QOpenSys, 'root' UDFS	*WX
		*FLR	QDLS	*RWX
		*DDIR	QOPT[11]	*WX
	Source Optical volume	*DDIR	QOPT[8]	*USE
	Target Optical volume	*DDIR	QOPT[8]	*CHANGE
CPY (continued)	Parent directory of origin object	*DIR[18]	QOpenSys, 'root' UDFS	*X
		*FLR	QDLS	*X
		Others	QSYS.LIB	*RX
		*DDIR	QOPT[11]	*X
	Path prefix (target destination)	*LIB[18]	QSYS.LIB	*WX
		*DIR[18]	QOpenSys, 'root' UDFS	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
	Path prefix (origin object)	*DDIR	QOPT[11]	*X
		*CHRSF		*IOSYSCNFG
CRTDIR	Parent directory	*DIR[18]	QOpenSys, 'root' UDFS	*WX
		*FLR	QDLS	*CHANGE
		*FILE	QSYS.LIB	*RX, *ADD
		Any		*ADD
		*DDIR	QOPT[11]	*WX

Command	Referenced Object	Object Type	File System	Authority Needed for Object[1]
CRTDIR (continued)	Path prefix	See General Rules on page 311		
	Optical volume	*DDIR	QOPT[8]	*CHANGE
CVTDIR (Q)[16]				
DSPAUT	Object	All	QDLS	*ALL
		All	All others	*OBJMGT or ownership
		ALL	QOPT[11]	None
	Optical volume	*DDIR	QOPT[8]	*USE
	Path prefix	See General Rules on page 311		
DSPCURDIR	Path prefix	*DIR[18]	QOpenSys, 'root' UDFS	*RX
		*FLR	QDLS	*RX
		*LIB[18], *FILE	QSYS.LIB	*RX
		*DIR[18]		*R
		*DDIR	QOPT[11]	*RX
DSPCURDIR (continued)	Current directory	*DIR[18]	QOpenSys, 'root' UDFS	*X
		*LIB[18], *FILE	QSYS.LIB	*X
		*FLR	QDLS	*X
		*DIR[18]		*R
		*DDIR	QOPT[11]	*X
	Optical volume	*DDIR*	QOPT[8]	*USE
DSPLNK	Any	Any	'root', QOpenSys, UDFS QSYS.LIB, QDLS, QOPT[11]	None
	File, Option 12 (Display Links)	*STMF, *SYMLNK, *DIR[18],*BLKSF, *SOCKET	'root', QOpenSys, UDFS	*R
DSPLNK (continued)	Symbolic link object	*SYMLNK	'root', QOpenSys, UDFS	None
	Optical volume	*DDIR	QOPT[8]	*USE
	Parent directory of referenced object - No Pattern [13]	*DIR[18]	'root', QOpenSys, UDFS	*X
		*LIB[18] *FILE	QSYS.LIB	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R

Command	Referenced Object	Object Type	File System	Authority Needed for Object[1]
DSPLNK (continued)	Parent directory of referenced object - Pattern specified[13]	*DIR[18]	'root', QOpenSys, UDFS	*R
		*LIB[18] *FILE	QSYS.LIB	*R
		*FLR	QDLS	*R
		*DDIR	QOPT[11]	*R
		*DDIR		*R
	Parent directory of referenced object- Option 8 (Display Attributes)	*DIR[18]	'root', QOpenSys, UDFS	*X
		*LIB[18] *FILE	QSYS.LIB	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R
DSPLNK (continued)	Parent directory of referenced object - Option 12 (Display Links)	*DIR[18]	'root', QOpenSys, UDFS	*RX
		*SYMLNK	'root', QOpenSys, UDFS	*X
		*LIB[18] *FILE	QSYS.LIB	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R
DSPLNK (continued)	Prefix of parent referenced object - No Pattern [13]	*DIR[18]	'root', QOpenSys, UDFS	*X
		*LIB[18] *FILE	QSYS.LIB	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R
DSPLNK (continued)	Prefix of parent referenced object - Pattern specified[13]	*DIR[18]	'root', QOpenSys, UDFS	*X
		*LIB[18] *FILE	QSYS.LIB	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R

Command	Referenced Object	Object Type	File System	Authority Needed for Object[1]
DSPLNK (continued)	Prefix of parent referenced object - Option 8 (Display Attributes)	*DIR[18]	'root', QOpenSys, UDFS	*RX
		*LIB[18] *FILE	QSYS.LIB	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R
DSPLNK (continued)	Prefix of parent referenced object - Option 12 (Display Links)	*DIR[18]	'root', QOpenSys, UDFS	*RX
		*SYMLNK	'root', QOpenSys, UDFS	*X
		*LIB[18] *FILE	QSYS.LIB	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R
DSPLNK (continued)	Relative Path Name [14] : Current working directory containing object -No Pattern[13]	*DIR[18]	'root', QOpenSys, UDFS	*RX
		*LIB[18] *FILE	QSYS.LIB	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*RX
		*DDIR		*R
	Relative Path Name [14] : Current working directory containing object -Pattern Specified[13]	*DIR[18]	'root', QOpenSys, UDFS	*RX
		*LIB[18] *FILE	QSYS.LIB	*RX
		*FLR	QDLS	*RX
		*DDIR	QOPT[11]	*RX
		*DDIR		*R
DSPLNK (continued)	Relative Path Name [14] : Prefix of current working directory containing object -No Pattern [13]	*DIR[18]	'root', QOpenSys, UDFS	*RX
		*LIB[18] *FILE	QSYS.LIB	*RX
		*FLR	QDLS	*RX
		*DDIR	QOPT[11]	*RX
		*DDIR		*R

Integrated File System Commands

Command	Referenced Object	Object Type	File System	Authority Needed for Object[1]
DSPLNK (continued)	Relative Path Name [14] : Prefix of current working directory containing object -Pattern specified[13]	*DIR[18]	'root', QOpenSys, UDFS	*RX
		*LIB[18] *FILE	QSYS.LIB	*RX
		*FLR	QDLS	*RX
		*DDIR	QOPT[11]	*RX
		*DDIR		*R
ENDJRN	Object	*DIR if Subtree(*ALL)	QOpenSys, 'root', UDFS	*R, *X, *OBJMGT
		*DIR if Subtree (*NONE), *SYMLNK, *STMF	QOpenSys, 'root'. UDFS	*R, *OBJMGT
		*DTAARA[18], *DTAQ[18]	QSYS.LIB	*OBJOPR, *READ, *OBJMGT
	Parent Directory	*DIR[18]	QOpenSys, 'root', UDFS	*X
		*LIB[18]	QSYS.LIB	*X
	Path Prefix	See General Rules on page 311		
MOV	Object moved within same file system	*DIR[18]	QOpenSys, 'root'	*OBJMGT, *W
		not *DIR	QOpenSys, 'root'	*OBJMGT
		*DOC	QDLS	*ALL
		*FILE	QSYS.LIB	*OBJOPR, *OBJMGT
		*MBR	QSYS.LIB	None
		other	QSYS.LIB	None
		*STMF	QOPT[11]	*W

Command	Referenced Object	Object Type	File System	Authority Needed for Object[1]
MOV (continued)	Path (source), parent directory	*DIR[18]	QOpenSys, 'root'	*WX
		*FLR	QDLS	*RWX
		*FILE	QSYS.LIB, 'root'	*RX, *OBJEXIST
		others	QOpenSys, 'root'	*RWX
	Path (target), parent directory	*DIR[18]	QSYS.LIB	*WX
		*FLR	QDLS	*CHANGE (*RWX)
		*FILE	QSYS.LIB	*X, *ADD, *DLT, *OBJMGT
		*LIB[18]	QSYS.LIB	*RWX
		*DDIR	QOPT[11]	*WX
MOV (continued)	Path prefix (target)	*LIB[18]	QSYS.LIB	*X, *ADD
		*FLR	QDLS	*X
		*DIR[18]	others	*X
		*DDIR	QOPT[11]	*X
	Object moved across file systems into QOpenSys, root or QDLS (stream file *STMF and *DOC, *MBR only) .	*STMF	QOpenSys, 'root' UDFS	*R, *OBJEXIST, *OBJMGT
		*DOC	QDLS	*ALL
		*MBR	QSYS.LIB	N/A
		*DSTMF	QOPT[11]	*RW
MOV (continued)	Moved into QSYS *MBR	*STMF	QOpenSys, 'root' UDFS	*R, *OBJMGT, *OBJEXIST
		*DOC	QDLS	*ALL
		*DSTMF	QOPT[11]	*RW
MOV (continued)	Path (source) moved across file systems, parent directory	*DIR[18]	QOpenSys, 'root' UDFS	*WX
		*FLR	QDLS	*X
		*FILE	QSYS. LIB	ownership, *RX, *OBJEXIST
		*DDIR	QOPT[11]	*WX
	Path Prefix	See General Rules on page 311		
	Optical volume (Source and Target)	*DDIR	QOPT[8]	*CHANGE

Integrated File System Commands

Command	Referenced Object	Object Type	File System	Authority Needed for Object[1]
RMVDIR	Directory	*DIR[18]	QOpenSys, 'root' UDFS	*OBJEXIST
		*LIB[18]	QSYS.LIB	*RX, *OBJEXIST
		*FILE	QSYS.LIB	*OBJOPR, *OBJEXIST
		*FLR	QDLS	*ALL
		*DDIR	QOPT[11]	*W
RMVDIR (continued)	Parent directory	*DIR[18]	QOpenSys, 'root' UDFS	*WX
		*FLR	QDLS	*X
		*LIB[18], *FILE	QSYS.LIB	*X
		*DDIR	QOPT[11]	*WX
	Path Prefix	See General Rules on page 311		
	Optical volume	*DDIR	QOPT[8]	*CHANGE
RMVLNK	Object	*DOC	QDLS	*ALL
		*MBR	QSYS.LIB	
		*FILE	QSYS.LIB	*OBJOPR, *OBJEXIST
		*JRNRCV[18]	QSYS.LIB	*OBJEXIST, *R
		other	QSYS.LIB	*OBJEXIST
		*DSTMF	QOPT[11]	*W
		any	QOpenSys, 'root' UDFS	*OBJEXIST
RMVLNK (continued)	Parent Directory	*FLR	QDLS	*X
		*FILE	QSYS.LIB	*X, *OBJEXIST
		*LIB[18]	QSYS.LIB	*X
		*DIR[18]	QOpenSys, 'root' UDFS	*WX
		*DDIR	QOPT[11]	*WX
	Path prefix	(See General Rules on page 311)		
	Optical volume	*DDIR	QOPT[8]	*CHANGE

Command	Referenced Object	Object Type	File System	Authority Needed for Object[1]
RNM	Object	*DIR[18]	QOpenSys, 'root' UDFS	*OBJMGT, *W
		Not *DIR	QOpenSys, 'root' UDFS	*OBJMGT
		*DOC, *FLR	QDLS	*ALL
		*MBR	QSYS.LIB	N/A
		*FILE	QSYS.LIB	*OBJMGT, *OBJOPR
		others	QSYS.LIB	*OBJMGT
		*DSTMF	QOPT[11]	*W
	Optical Volume (Source and Target)	*DDIR	QOPT[8]	*CHANGE
RNM (continued)	Parent directory	*DIR[18]	QOpenSys, 'root' UDFS	*WX
		*FLR	QDLS	*CHANGE (*RWX)
		*FILE	QSYS.LIB	*X, *OBJMGT
		*LIB[18]	QSYS.LIB	*X,*UPD
		*DDIR	QOPT[11]	*WX
	Path prefix	*LIB[18]	QSYS.LIB	*X, *UPD
		any	QOpenSys, 'root' user defined file systemQDLS	*X
RST (Q)	Object, if it exists[2]	Any	QOpenSys, 'root' UDFS	*W, *OBJEXIST
			QSYS.LIB	Varies [10]
			QDLS	*ALL
				*OBJMGT, *OBJALTER, *READ, *UPD
	Path prefix	See General Rules on page 311		
RST (Q) (continued)	Parent directory of object being restored[2]	*DIR[18]	QOpenSys, 'root' UDFS	*WX
	Parent directory of object being restored, if the object does not exist[2]	*FLR	QDLS	*CHANGE
		*DIR[18]		*OBJMGT, *OBJALTER, *READ, *ADD, *UPD
	User profile owning new object being restored[2]	*USRPRF	QSYS.LIB	*ADD
	Tape unit, diskette unit, optical unit, or save file	*DEVD, *FILE	QSYS.LIB	*RX

Integrated File System Commands

Command	Referenced Object	Object Type	File System	Authority Needed for Object[1]
RST (Q) (continued)	Library for device description or save file	*LIB[18]	QSYS.LIB	*EXECUTE
	Output file, if specified	*STMF	QOpenSys, 'root' UDFS	*W
		*USRSPC[18]	QSYS.LIB	*RWX
	Path prefix of output file	*DIR[18]	QOpenSys, 'root' UDFS	*X
		*LIB[18]	QSYS.LIB	*RX
RST (Q) (continued)	Optical volume if restoring from optical device	*DDIR	QOPT[8]	*USE
	Optical path prefix and parent if restoring from optical device	*DDIR	QOPT[11]	*X
	Optical file if restoring from optical device	*DSTMF	QOPT[11]	*R
RTVCURDIR	Path prefix	*DIR[18]	QOpenSys, 'root', UDFS,QDLS, QOPT[11]	*RX
		*DDIR	QOPT[11]	*RX
		*FLR	QDLS	*RX
		*LIB[18], *FILE	QSYS.LIB	*RX
		Any		*R
RTVCURDIR (continued)	Current directory	*DIR[18]	QOpenSys, 'root', UDFS,QOPT[11]	*X
		*DDIR	QOPT[11]	*X
		*LIB[18], *FILE	QSYS.LIB	*X
		*FLR	QDLS	*X
		Any		*R
SAV	Object[2]	Any	QOpenSys, 'root' UDFS	*R, *OBJEXIST
			QSYS.LIB	Varies [10]
			QDLS	*ALL
				*OBJMGT, *R
	Path prefix	See General Rules on page 311		
	Tape unit, diskette unit, or optical unit	*DEVD	QSYS.LIB	*RX
SAV (continued)	Save file, if empty	*FILE	QSYS.LIB	*USE, *ADD
	Save file, if not empty	*FILE	QSYS.LIB	*OBJMGT, *USE, *ADD
	Save-while-active message queue	*MSGQ	QSYS.LIB	*OBJOPR, *ADD
	Libraries for device description, save file, save-while-active message queue	*LIB[18]	QSYS.LIB	*EXECUTE

Command	Referenced Object	Object Type	File System	Authority Needed for Object[1]
SAV (continued)	Output file, if specified	*STMF	QOpenSys, 'root' UDFS	*W
		*USRSPC[18]	QSYS.LIB	*RWX
	Path prefix of output file	*DIR[18]	QOpenSys, 'root' UDFS	*X
		*LIB[18]	QSYS.LIB	*RX
SAV (continued)	Optical volume, if saving to optical device	*DDIR	QOPT[8]	*CHANGE
	Optical path prefix if saving to optical device	*DDIR	QOPT[11]	*X
	Optical parent directory if saving to optical device	*DDIR	QOPT[11]	*WX
	Optical file (If it previously exists)	*DSTMF	QOPT[11]	*RW
SAVRST	On the source system, same authority as required by SAV command.			
	On the target system, same authority as required by RST command.			
STRJRN	Object	*DIR if Subtree(*ALL)	QOpenSys, 'root', UDFS	*R, *X, *OBJMGT
		*DIR if subtree(*NONE), *SYMLNK, *STMF	QOpenSys, 'root', UDFS	*R, *OBJMGT
		*DTAARA[18], *DTAQ	QSYS.LIB	*OBJOPR, *READ, *OBJMGT
	Parent Directory	*DIR[18]	QOpenSys, 'root', UDFS	*X
		*LIB[18]	QSYS.LIB	*X
	Path Prefix	See General Rules on page 311		
WRKAUT[6, 7]	Object	*DOC or *FLR	QDLS	*ALL
		All	not QDLS	*OBJMGT or ownership
		*DDIR and *DSTMF	QOPT[11]	*NONE
	Path prefix	See General Rules on page 311		
	Optical volume	*DDIR	QOPT[8]	*USE

Integrated File System Commands

Command	Referenced Object	Object Type	File System	Authority Needed for Object[1]
WRKLNK	Any	Any	'root', QOpenSys, UDFS, QSYS.LIB, QDLS, QOPT[11]	None
	File, Option 12 (Display Links)	*STMF, *SYMLNK, *DIR[18], *BLKSF, *SOCKET	'root', QOpenSys, UDFS	*R
	Symbolic link object	*SYMLNK	'root', QOpenSys, UDFS	None
	Optical volume	*DDIR	QOPT[8]	*USE
WRKLNK (continued)	Parent directory of referenced object - No Pattern [13]	*DIR[18]	'root', QOpenSys, UDFS	*X
		*LIB[18] *FILE	QSYS.LIB	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R
WRKLNK (continued)	Parent directory of referenced object - Pattern Specified	*DIR[18]	'root', QOpenSys, UDFS	*R
		*LIB[18] *FILE	QSYS.LIB	*R
		*FLR	QDLS	*R
		*DDIR	QOPT[11]	*R
		*DDIR		*R
WRKLNK (continued)	Parent directory of referenced object- Option 8 (Display Attributes)	*DIR[18]	'root', QOpenSys, UDFS	*X
		*LIB[18] *FILE	QSYS.LIB	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R
WRKLNK (continued)	Parent directory of referenced object - Option 12 (Display Links)	*DIR[18]	'root', QOpenSys, UDFS	*RX
		*SYMLNK	'root', QOpenSys, UDFS	*X
		*LIB[18] *FILE	QSYS.LIB	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R

Command	Referenced Object	Object Type	File System	Authority Needed for Object[1]
WRKLNK (continued)	Prefix of parent referenced object - No Pattern [13]	*DIR[18]	'root', QOpenSys, UDFS	*X
		*LIB[18] *FILE	QSYS.LIB	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R
WRKLNK (continued)	Prefix of parent referenced object - Pattern specified[13]	*DIR[18]	'root', QOpenSys, UDFS	*X
		*LIB[18] *FILE	QSYS.LIB	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R
WRKLNK (continued)	Prefix of parent referenced object - Option 8 (Display Attributes)	*DIR[18]	'root', QOpenSys, UDFS	*RX
		*LIB[18] *FILE	QSYS.LIB	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R
WRKLNK (continued)	Prefix of parent referenced object - Option 12 (Display Links)	*DIR[18]	'root', QOpenSys, UDFS	*RX
		*SYMLNK	'root', QOpenSys, UDFS	*X
		*LIB[18] *FILE	QSYS.LIB	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R

Integrated File System Commands

Command	Referenced Object	Object Type	File System	Authority Needed for Object[1]
WRKLNK (continued)	Relative Path Name [14,] : Current working directory containing object -No Pattern[13]	*DIR[18]	'root', QOpenSys, UDFS	*RX
		*LIB[18] *FILE	QSYS.LIB	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*RX
		*DDIR		*R
	Relative Path Name [14] : Current working directory containing object -Pattern Specified[13]	*DIR[18]	'root', QOpenSys, UDFS	*RX
		*LIB[18] *FILE	QSYS.LIB	*RX
		*FLR	QDLS	*RX
		*DDIR	QOPT[11]	*RX
		*DDIR		*R
WRKLNK (continued)	Relative Path Name [14] : Prefix of current working directory containing object -No Pattern [13]	*DIR[18]	'root', QOpenSys, UDFS	*RX
		*LIB[18] *FILE	QSYS.LIB	*RX
		*FLR	QDLS	*RX
		*DDIR	QOPT[11]	*RX
		*DDIR		*R
	Relative Path Name [14] Prefix of current working directory containing object -Pattern specified[13]	*DIR[18]	'root', QOpenSys, UDFS	*RX
		*LIB[18] *FILE	QSYS.LIB	*RX
		*FLR	QDLS	*RX
		*DDIR	QOPT[11]	*RX
		*DDIR		*R

[1] Adopted authority is not used for Integrated File system commands.

[2] If you have *SAVSYS special authority, you do not need the authority specified for the QSYS.LIB, QDLS, QOpenSys, and "root" file systems.

[3] The authority required varies by object type. See the description of the QLIRNMO API in the Information Center (see "Prerequisite and related information" on page xvi for details). If the object is a database member, see the authorities for the Rename Member (RNMM) command.

[4] You must have *AUDIT special authority to change an auditing value.

[5] If the user issuing the command does not have *ALLOBJ authority, the user must be a member of the new primary group.

[6] To use an individual operation, you must have the authority required by the operation

[7] These commands require the authority shown plus the authorities required for the DSPCURDIR command.

[8] Optical volumes are not actual system objects. The link between the optical volume and the authorization list used to secure the volume is maintained by the optical support function.

Command	Referenced Object	Object Type	File System	Authority Needed for Object[1]

[9] See Chapter 7 of iSeries Optical Support book for information on restrictions regarding this command.

[10] Authority required varies by the native command used. See the respective SAVOBJ or RSTOBJ command for the required authority.

[11] Authority required by QOPT against media formatted in ″Universal Disk Format″ (UDF).

[12] *ADD is needed only when object being moved to is a *MRB.

[13] Pattern: In some commands, an asterick (*) or a question mark (?) can be used in the last component of the path name to search for names matching a pattern.

[14] Relative path name: If a path name does not begin with a slash, the predecessor of the first component of the path name is taken to be the current working directory of the process. For example, if a path name of 'a/b' is specified, and the current working directory is '/home/john', then the object being accessed is '/home/john/a/b'.

[15] If you have *ALLOBJ special authority, you do not need the listed authority.

[16] You must have have *ALLOBJ special authority to use this command.

[17] In the above table, QSYS.LIB refers to independant ASP QSYS.LIB file systems as well as QSYS.LIB file system.

[18] This object is allowed on independant ASP.

Interactive Data Definition Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
ADDDTADFN	Data dictionary	*CHANGE	*EXECUTE
	File	*OBJOPR, *OBJMGT	*EXECUTE
CRTDTADCT	Data dictionary		*READ, *ADD
DLTDTADCT [3]	Data dictionary	OBJEXIST, *USE	
DSPDTADCT	Data dictionary	*USE	*EXECUTE
LNKDTADFN [1]	Data dictionary	*USE	*EXECUTE
	File	*OBJOPR, *OBJMGT	*EXECUTE
STRIDD			
WRKDTADCT [2]	Data dictionary	*OBJOPR	*EXECUTE
WRKDBFIDD [2]	Data dictionary	*USE [4]	*EXECUTE
	Database file	*OBJOPR	*EXECUTE
WRKDTADFN [1]	Data dictionary	*USE, *CHANGE	*EXECUTE

[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.

Internetwork Packet Exchange (IPX) Commands

Appendix C shows which IBM-supplied user profiles are authorized to the command. The security officer can grant *USE authority to others.

		Authority Needed	
Command	**Referenced Object**	**For Object**	**For Library**
DLTIPXD	IPX description	*OBJEXIST	*EXECUTE
DSPIPXD	IPX description	*USE	*EXECUTE
WRKIPXD	IPX description	*OBJOPR	*EXECUTE

Information Search Index Commands

		Authority Needed	
Command	**Referenced Object**	**For Object**	**For Library**
ADDSCHIDXE	Search index	*CHANGE	*USE
	Panel group	*USE	*EXECUTE
CHGSCHIDX	Search index	*CHANGE	*USE
CRTSCHIDX	Search Index		*READ, *ADD
DLTSCHIDX	Search index	*OBJEXIST	*EXECUTE
RMVSCHIDXE	Search index	*CHANGE	*USE
STRSCHIDX	Search index	*USE	*EXECUTE
WRKSCHIDX [1]	Search index	*ANY	*USE
WRKSCHIDXE	Search index	*USE	*USE

1 To use an individual operation, you must have the authority required by the operation .

IPL Attribute Commands

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.

These commands do not require authorities to objects:
CHGIPLA (Q) [1] DSPIPLA

1 To use this command, you must have *SECADM and *ALLOBJ special authorities.

Job Commands

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	For Library
BCHJOB	Job description [9,11]	*USE	*EXECUTE
	User profile in job description [10]	*USE	*EXECUTE
	Sort sequence table [10]	*USE	*EXECUTE
	Message queue [7]	*USE, *ADD	*EXECUTE
	Job queue [10,11]	*READ	*EXECUTE
	Output queue [7]	*USE	*EXECUTE
CHGACGCDE [1]			
CHGGRPA [4]	Message queue if associating a message queue with a group	*OBJOPR	*EXECUTE
CHGJOB [1,2,3]	New job queue, if changing the job queue[10,11]	*READ	*EXECUTE
	New output queue, if changing the output queue[7]	*READ	*EXECUTE
	Sort sequence table[7]	*USE	*EXECUTE
CHGPJ	User profile for the program start request to specify *PGMSTRRQS	*USE	*EXECUTE
	User profile and job description	*USE	*EXECUTE
CHGSYSJOB(Q) [13]			
CHGUSRTRC[14]	User trace buffer when CLEAR (*YES) is used.[15]	*OBJOPR	*EXECUTE
	User trace buffer when MAXSTG is used[15]	*CHANGE, *OBJMGT	*USE
	User trace buffer when TRCFULL is used. [15]	*OBJOPR	*EXECUTE
DLTUSRTRC	User trace buffer [15]	*OBJOPR, *OBJEXIST	*EXECUTE
DLYJOB [4]			
DMPUSRTRC	User trace buffer [15]	*OBJOPR	*EXECUTE
DSCJOB [1]			
DSPACTPJ			
DSPJOB [1]			
DSPJOBTBL			
DSPJOBLOG [1,5]	Outfile and member exist	*OBJOPR, *OBJMGT, *ADD	*EXECUTE
	Member does not exist	*OBJOPR, *OBJMGT, *ADD	*EXECUTE, *ADD
	Outfile does not exist	*OBJOPR	*EXECUTE, *ADD
ENDGRPJOB			
ENDJOB [1]			
ENDJOBABN [1]			
ENDPJ [6]			
HLDJOB [1]			
RLSJOB [1]			
RRTJOB			
RTVJOBA			

Job Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
SBMDBJOB	Database file	*USE	*EXECUTE
	Job queue	*READ	*EXECUTE
SBMDKTJOB	Message queue	*USE, *ADD	*EXECUTE
	Job queue and device description	*READ	*EXECUTE
SBMJOB [2, 12]	Job description [9,11]	*USE	*EXECUTE
	Libraries in the library list (system, current, and user)[7]	*USE	
	Message queue	*USE, *ADD	*EXECUTE
	User profile [10,11]	*USE	*EXECUTE
	User profile in job description [10]	*USE (at level 40)	*EXECUTE
	Job queue [10,11]	*READ	*EXECUTE
	Output queue [7]	*READ	*EXECUTE
	Sort sequence table [10]	*USE	*EXECUTE
SBMNETJOB	Database file	*USE	*EXECUTE
STRPJ [6]	Subsystem description	*USE	*EXECUTE
	Program	*USE	*EXECUTE
TFRBCHJOB	Job queue	*READ	*EXECUTE
TFRGRPJOB	Initial group program	*USE	*EXECUTE
TFRJOB [8]	Job queue	*READ	*EXECUTE
	Subsystem description to which the job queue is allocated	*USE	*EXECUTE
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. If you have *SPLCTL special authority, you do not need any authority to the job queue. However, you need authority to the library that contains the job queue.

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.

Command	Referenced Object	Authority Needed	
		For Object	For Library
[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] The adopted authority of the user issuing the CHGJOB or SBMJOB command is used.			
[12] You must be authorized to the user profile and the job description; the user profile must also be authorized to the job description.			
[13] To change certain job attributes, even in the user's own job, requires job control (*JOBCTL) and all object (*ALLOBJ) special authorities.			
[14] 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.			
[15] A user trace buffer is a user space (*USRSPC) object in library QUSRSYS by the name QPOZnnnnnn, where 'nnnnnn' is the job number of the job using the user trace facility.			

Job Description Commands

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	For Library
CHGJOBD	Job description	*OBJOPR, *OBJMGT	*EXECUTE
	User profile (USER)	*OBJOPR	*EXECUTE
CRTJOBD (Q)	User profile (USER)	*OBJOPR	*EXECUTE
	Job description		*READ, *ADD
DLTJOBD	Job description	*OBJEXIST	*EXECUTE
DSPJOBD	Job description	*OBJOPR, *READ	*EXECUTE
PRTJOBDAUT [1]			
WRKJOBD	Job description	Any	*USE
[1] You must have *ALLOBJ special authority to use this command.			

Job Queue Commands

Command	Referenced Object	Authority Needed		Job Queue Parameters [4]		Special Authority
		For Object	For Library	AUTCHK	OPRCTL	
CLRJOBQ [1]	Job queue	*READ, *ADD, *DLT	*EXECUTE	*DTAAUT		
		Owner [2]	*EXECUTE	*OWNER		
			*EXECUTE		*YES	*JOBCTL
CRTJOBQ [1]	Job queue		*READ, *ADD			
DLTJOBQ	Job queue	*OBJEXIST	*EXECUTE			
HLDJOBQ [1]	Job queue	*READ, *ADD, *DLT	*EXECUTE	*DTAAUT		
		Owner [2]	*EXECUTE	*OWNER		
			*EXECUTE		*YES	*JOBCTL
PRTQAUT [5]						
RLSJOBQ [1]	Job queue	*READ, *ADD, *DLT	*EXECUTE	*DTAAUT		
		Owner [2]	*EXECUTE	*OWNER		
			*EXECUTE		*YES	*JOBCTL
WRKJOBQ [1,3]	Job queue	*READ	*EXECUTE	*DTAAUT		
		Owner [2]	*EXECUTE	*OWNER		
			*EXECUTE		*YES	*JOBCTL

1 If you have *SPLCTL special authority, you do not need any authority to the job queue but you need authority to the library containing 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 *EXECUTE authority.

4 To display the job queue parameters, use the QSPRJOBQ API.

5 You must have *ALLOBJ special authority.

Job Schedule Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
ADDJOBSCDE	Job schedule	*CHANGE	*EXECUTE
	Job description [1]	*USE	*EXECUTE
	Job queue [1,2]	*READ	*EXECUTE
	User profile	*USE	*EXECUTE
	Message queue [1]	*USE, *ADD	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGJOBSCDE [3]	Job schedule	*CHANGE	*EXECUTE
	Job description [1]	*USE	*EXECUTE
	Job queue [1,2]	*READ	*EXECUTE
	User profile	*USE	*EXECUTE
	Message queue [1]	*USE, *ADD	*EXECUTE
HLDJOBSCDE [3]	Job schedule	*CHANGE	*EXECUTE
RLSJOBSCDE [3]	Job schedule	*CHANGE	*EXECUTE
RMVJOBSCDE [3]	Job schedule	*CHANGE	*EXECUTE
WRKJOBSCDE [4]	Job schedule	*USE	*EXECUTE

[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.

Journal Commands

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	For Library or Directory
ADDRMTJRN	Source journal	*CHANGE, *OBJMGT	*EXECUTE
	Target journal		*EXEC,*ADD
APYJRNCHG (Q)	Journal	*USE	*EXECUTE
	Journal receiver	*USE	*EXECUTE
	Non-IFS objects whose journaled changes are being applied	*OBJMGT, *CHANGE	*EXECUTE
	IFS objects whose journal changes are being applied	*RW, *OBJMGT	*RX (if subtree *ALL)
CHGJRN (Q)	Journal receiver, if specified	*OBJMGT, *USE	*EXECUTE
	Attached journal receiver	*OBJMGT, *USE	*EXECUTE
	Journal	*OBJOPR, *OBJMGT, *UPD	*EXECUTE
	Journal if RCVSIZOPT(*MINFIXLEN) is specified.	*OBJOPR, *OBJMGT, *UPD, *OBJALTER	*EXECUTE
CHGRMTJRN	Source journal	*CHANGE, *OBJMGT	*EXECUTE
	Source journal	*USE, *OBJMGT	*EXECUTE

Journal Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library or Directory
CMPJRNIMG	Journal	*USE	*EXECUTE
	Journal receiver	*USE	*EXECUTE
	File	*USE	*EXECUTE
CRTJRN	Journal		*READ, *ADD
	Journal receiver	*OBJOPR, *OBJMGT, *READ	*EXECUTE
DLTJRN	Journal	*OBJOPR, *OBJEXIST	*EXECUTE
DSPAUDJRNE (Q)[8]			
DSPJRN [6]	Journal	*USE	*EXECUTE
	Journal if FILE(*ALLFILE) is specified, the specified file has been deleted from the system or *IGNFILSLT is specified for any selected journal codes or the journal is a remote journal.	*OBJEXIST, *USE	*EXECUTE
	Journal receiver	*USE	*EXECUTE
	File if specified	*USE	*EXECUTE
	Output file	See General Rules on page 311	See General Rules on page 311
DSPJRNMNU[1]			
ENDJRN	See "Integrated File System Commands" on page 351		
ENDJRNAP	Journal	*OBJOPR, *OBJMGT	*EXECUTE
	File	*OBJOPR, *OBJMGT	*EXECUTE
ENDJRNOBJ	Journal	*OBJOPR, *OBJMGT	*EXECUTE
	Object	*OBJOPR, *READ, *OBJMGT	*EXECUTE
ENDJRNPF	Journal	*OBJOPR, *OBJMGT	*EXECUTE
	File	*OBJOPR, *OBJMGT, *READ	*EXECUTE
JRNAP [2]			
JRNPF [3]			
RCVJRNE	Journal	*USE	*EXECUTE
	Journal if FILE(*ALLFILE) is specified, the specified file has been deleted from the system or *IGNFILSLT is specified for any selected journal codes or the journal is a remote journal.	*OBJEXIST, *USE	*EXECUTE
	Journal receiver	*USE	*EXECUTE
	File	*USE	*EXECUTE
	Exit program	*EXECUTE	*EXECUTE
RMVJRNCHG (Q)	Journal	*USE	*EXECUTE
	Journal receiver	*USE	*EXECUTE
	Non-IFS objects whose journaled changes are being removed	*OBJMGT, *CHANGE	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library or Directory
RTVJRNE	Journal	*USE	*EXECUTE
	Journal if FILE(*ALLFILE) is specified, the specified file has been deleted from the system or *IGNFILSLT is specified for any selected journal codes or the journal is a remote journal.	*OBJEXIST, *USE	*EXECUTE
	Journal receiver	*USE	*EXECUTE
	File	*USE	*EXECUTE
SNDJRNE	Journal	*OBJOPR, *ADD	*EXECUTE
	Non-IFS object if specified	*OBJOPR	*EXECUTE
	IFS object if specified	*R	*X
STRJRN	See "Integrated File System Commands" on page 351		
STRJRNAP	Journal	*OBJOPR, *OBJMGT	*EXECUTE
	File	*OBJOPR, *OBJMGT	*EXECUTE
STRJRNPF	Journal	*OBJOPR, *OBJMGT	*EXECUTE
	File	*OBJOPR, *OBJMGT	*EXECUTE
STRJRNOBJ	Journal	*OBJOPR, *OBJMGT	*EXECUTE
	Object	*OBJOPR, *READ, *OBJMGT	*EXECUTE
WRKJRN [4] (Q)	Journal	*USE	*READ[7]
	Journal receiver if receiver information is requested	*USE	*EXECUTE
	File if forward or backout recovery is requested	*OBJMGT, *CHANGE	*EXECUTE
	Objects that are deleted during recovery	*OBJEXIST	*EXECUTE
WRKJRNA [6]	Journal	*OBJOPR and a data authority other than *EXECUTE	*EXECUTE
	Journal receiver [5]	*OBJOPR and a data authority other than *EXECUTE	*EXECUTE

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 you must have the authority required for the RSTOBJ command.

5 *OBJOPR and *OBJEXIST authority is required for journal receivers if the option is chosen to delete receivers.

6 To specify JRN(*INTSYSJRN), you must have *ALLOBJ special authority.

7 *READ authority to the journal's library is required to display the WRKJRN menu. *EXECUTE authority to the library is required to use an option on the menu.

8 You must have *ALLOBJ and *AUDIT special authorities to use this command.

Journal Receiver Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTJRNRCV	Journal receiver		*READ, *ADD
DLTJRNRCV	Journal receiver	*OBJOPR, *OBJEXIST, and a data authority other than *EXECUTE	*EXECUTE
	Journal	*OBJOPR	*EXECUTE
DSPJRNRCVA	Journal receiver	*OBJOPR and a data authority other than *EXECUTE	*EXECUTE
	Journal, if attached	*OBJOPR	*EXECUTE
WRKJRNRCV [1], [2], [3]	Journal receiver	Any authority	*USE

[1] To use an individual operation, you must have the authority required by the operation .

[2] *OBJOPR and *OBJEXIST authority is required for journal receivers if the option is chosen to delete receivers.

[3] *OBJOPR and a data authroity other than *EXECUTE is required for journal receivers if the option is chosen to display the description.

Language Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTBNDC	Source file	*USE	*EXECUTE
	Externally described device files and database files referred to in source program	*OBJOPR	*EXECUTE
	Program: REPLACE(*NO)		*READ, *ADD
	Program: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	Directory specified in OUTPUT, PPSRCSTMF or MAKEDEP parameter	*USE	*EXECUTE
	File specified in OUTPUT, PPSRCSTMF or MAKEDEP parameter	See General Rules on page 311	*READ, *ADD
CRTBNDCBL	Source file	*USE	*EXECUTE
	Externally described device files and database files referred to in source program	*OBJOPR	*EXECUTE
	Program: REPLACE(*NO)		*READ,*ADD
	Program: REPLACE(*YES)	See General Rules on page 311	*READ,*ADD
	Binding directory	*USE	*EXECUTE
	Table specified in SRTSEQ parameter	*USE	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTBNDCL	Source file	*USE	*EXECUTE
	Externally described device files and database files referred to in source program	*OBJOPR	*EXECUTE
	Program: REPLACE(*NO)		*READ, *ADD
	Program: REPLACE(*YES)	See General Rules on page 311	See General Rules on page 311
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
CRTBNDCPP	Source File	*USE	*EXECUTE
	Externally described device files and database files referred to in source program	*OBJOPR	*EXECUTE
	Program: REPLACE(*NO)		*READ, *ADD
	Program: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	Directory specified in OUTPUT, PPSRCSTMF, TEMPLATE or MAKEDEP parameter	*USE	*EXECUTE
	File specified in OUTPUT, PPSRCSTMF, TEMPLATE or MAKEDEP parameter	See General Rules on page 311	*READ, *ADD
	Headers generated by TEMPLATE parameter	*USE	*EXECUTE
CRTBNDRPG	Source file	*USE	*EXECUTE
	Externally described device files and database files referred to in source program	*OBJOPR	*EXECUTE
	Program: REPLACE(*NO)		*READ,*ADD
	Program: REPLACE(*YES)	See General Rules on page 311	*READ,*ADD
	Binding directory	*USE	*EXECUTE
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
CRTCBLMOD	Source file	*USE	*EXECUTE
	Externally described device files and database files referred to in source program	*OBJOPR	*EXECUTE
	Module: REPLACE(*NO)		*READ, *ADD
	Module: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
CRTCLD	Source file	*USE	*EXECUTE
	Locale object - REPLACE(*NO)		*READ, *ADD
	Locale object - REPLACE(*YES)	See General Rules on page 311	*READ, *ADD

Language Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTCLMOD	Source file	*USE	*EXECUTE
	Externally described device files and database files referred to in source program	*OBJOPR	*EXECUTE
	Program: REPLACE(*NO)		*READ, *ADD
	Program: REPLACE(*YES)	See General Rules on page 311	See General Rules on page 311
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
CRTCLPGM	Source file	*USE	*EXECUTE
	Externally described device files and database files referred to in source program	*OBJOPR	*EXECUTE
	Program: REPLACE(*NO)		*READ, *ADD
	Program: REPLACE(*YES)	See General Rules on page 311	See General Rules on page 311
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
CRTCBLPGM (COBOL/400* licensed program or S/38 environment)	Source file	*USE	*EXECUTE
	Externally described device files and database files referred to in source program	*OBJOPR	*EXECUTE
	Program: REPLACE(*NO)		*READ, *ADD
	Program: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
CRTCMOD	Source file	*USE	*EXECUTE
	Externally described device files and database files referred to in source program	*OBJOPR	*EXECUTE
	Module: REPLACE(*NO)		*READ, *ADD
	Module: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	File specified in OUTPUT, PPSRCSTMF or MAKEDEP parameter	*USE	*EXECUTE
	File specified in OUTPUT, PPSRCSTMF or MAKEDEP parameter	See General Rules on page 311	*READ, *ADD
CRTCPPMOD	Source file	*USE	*EXECUTE
	Externally described device files and database files referred to in source program	*OBJOPR	*EXECUTE
	Module: REPLACE(*NO)		*READ, *ADD
	Module: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	Directory specified in OUTPUT, PPSRCSTMF, TEMPLATE or MAKEDEP parameter	*USE	*EXECUTE
	File specified in OUTPUT, PPSRCSTMF, TEMPLATE or MAKEDEP parameter	See General Rules on page 311	*READ, *ADD
	Headers generated by TEMPLATE parameter	*USE	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTRPGMOD	Source file	*USE	*EXECUTE
	Externally described device files and database files referred to in source program	*OBJOPR	*EXECUTE
	Module: REPLACE(*NO)		*READ,*ADD
	Module: REPLACE(*YES)	See General Rules on page 311	*READ,*ADD
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
CRTRPGPGM (RPG/400* licensed program and S/38 environment)	Source file	*USE	*EXECUTE
	Externally described device files and database files referred to in source program	*OBJOPR	*EXECUTE
	Program: REPLACE(*NO)		*READ, *ADD
	Program: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
CRTRPTPGM (RPG/400 licensed program and S/38 environment)	Source file	*USE	*EXECUTE
	Program - REPLACE(*NO)		*READ, *ADD
	Program - REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	Source file for generated RPG program	See General Rules for replacing and adding members on page 311	See General Rules for replacing and adding members on page 311
	Externally described device files and database files referred to in source program	*OBJOPR	*EXECUTE
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
CRTS36CBL (S/36 environment)	Source file	*USE	*EXECUTE
	Program: REPLACE(*NO)		*READ, *ADD
	Program: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
CRTS36RPG	Source file	*USE	*READ, *ADD
	Program: REPLACE(*NO)		*READ, *ADD
	Program - REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
CRTS36RPGR	Source file	*USE	*READ, *ADD
	Display file: REPLACE(*NO)		*READ, *ADD
	Display file: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
CRTS36RPT	Source file	*USE	*EXECUTE
	Source file for generated RPG program	See General Rules for replacing and adding members on page 311	See General Rules for replacing and adding members on page 311
	Program: REPLACE(*NO)		*READ, *ADD
	Program: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD

Language Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTSQLC OS/400' (DB2 Query Manager and SQL Development for OS/400 licensed program) [1]	Source file	*OBJOPR, *READ	*EXECUTE
	To Source file	*OBJOPR, *OBJMGT, *EXIST, *READ, *ADD, *UPDATE, *DELETE, *EXECUTE	*ADD, *EXECUTE
	Data description specifications	*OBJOPR	*EXECUTE
	Program: REPLACE(*NO)		*READ, *ADD
	Program: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
CRTSQLCI (DB2 Query Manager and SQL Development for OS/400 licensed program) [1]	Source file	*OBJOPR, *READ	*EXECUTE
	To Source file	*OBJOPR, *OBJMGT, *EXIST, *READ, *ADD, *UPDATE, *DELETE, *EXECUTE	*ADD, *EXECUTE
	Data description specifications	*OBJOPR	*EXECUTE
	Object: REPLACE(*NO)		*READ, *ADD
	Object: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
CRTSQLCBL (DB2 Query Manager and SQL Development for OS/400 licensed program) [1]	Source file	*OBJOPR, *READ	*EXECUTE
	To Source file	*OBJOPR, *OBJMGT, *EXIST, *READ, *ADD, *UPDATE, *DELETE, *EXECUTE	*ADD, *EXECUTE
	Data description specifications	*OBJOPR	*EXECUTE
	Program: REPLACE(*NO)		*READ, *ADD
	Program: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
CRTSQLCBLI (DB2 Query Manager and SQL Development for OS/400 licensed program) [1]	Source file	*OBJOPR, *READ	*EXECUTE
	To Source file	*OBJOPR, *OBJMGT, *EXIST, *READ, *ADD, *UPDATE, *DELETE, *EXECUTE	*ADD, *EXECUTE
	Data description specifications	*OBJOPR	*EXECUTE
	Object: REPLACE(*NO)		*READ, *ADD
	Object: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	Table specified in SRTSEQ parameter	*USE	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTSQLCPPI (DB2 Query Manager and SQL Development for OS/400 licensed program) [1]	Source file	*OBJOPR, *READ	*EXECUTE
	To Source file	*OBJOPR, *OBJMGT, *EXIST, *READ, *ADD, *UPDATE, *DELETE, *EXECUTE	*ADD, *EXECUTE
	Data description specifications	*OBJOPR	*EXECUTE
	Program: REPLACE(*NO)		*READ, *ADD
	Program: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
CRTSQLFTN (DB2 Query Manager and SQL Development for OS/400 licensed program) [1]	Source file	*OBJOPR, *READ	*EXECUTE
	To Source file	*OBJOPR, *OBJMGT, *EXIST, *READ, *ADD, *UPDATE, *DELETE, *EXECUTE	*ADD, *EXECUTE
	Data description specifications	*OBJOPR	*EXECUTE
	Program: REPLACE(*NO)		*READ, *ADD
	Program: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
CRTSQLPLI (DB2 Query Manager and SQL Development for OS/400 licensed program) [1]	Source file	*OBJOPR, *READ	*EXECUTE
	To Source file	*OBJOPR, *OBJMGT, *EXIST, *READ, *ADD, *UPDATE, *DELETE, *EXECUTE	*ADD, *EXECUTE
	Data description specifications	*OBJOPR	*EXECUTE
	Program: REPLACE(*NO)		*READ, *ADD
	Program: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
CRTSQLRPG (DB2 Query Manager and SQL Development for OS/400 licensed program) [1]	Source file	*OBJOPR, *READ	*EXECUTE
	To Source file	*OBJOPR, *OBJMGT, *EXIST, *READ, *ADD, *UPDATE, *DELETE, *EXECUTE	*ADD, *EXECUTE
	Data description specifications	*OBJOPR	*EXECUTE
	Program: REPLACE(*NO)		*READ, *ADD
	Program: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	Table specified in SRTSEQ parameter	*USE	*EXECUTE

Language Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTSQLRPGI (DB2 Query Manager and SQL Development for OS/400 licensed program) [1]	Source file	*OBJOPR, *READ	*EXECUTE
	To Source file	*OBJOPR, *OBJMGT, *EXIST, *READ, *ADD, *UPDATE, *DELETE, *EXECUTE	*ADD, *EXECUTE
	Data description specifications	*OBJOPR	*EXECUTE
	Object: REPLACE(*NO)		*READ, *ADD
	Object: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
CVTRPGSRC	Source file	*USE	*EXECUTE
	Output file	*OBJOPR, *OBJMGT, *ADD	*EXECUTE
	Log file	*OBJOPR, *OBJMGT, *ADD	*EXECUTE
CVTSQLCPP [1]	Source file	*OBJOPR, *READ	*EXECUTE
	To Source file	*OBJOPR, *OBJMGT, *EXIST, *READ, *ADD, *UPDATE, *DELETE, *EXECUTE	*ADD, *EXECUTE
	Data description specifications	*OBJOPR	*EXECUTE
	Program: REPLACE(*NO)		*READ, *ADD
	Program: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
ENDCBLDBG (COBOL/400 licensed program or S/38 environment)	Program	*CHANGE	*EXECUTE
ENTCBLDBG (S/38 environment)	Program	*CHANGE	*EXECUTE
DLTCLD	Locale object	*OBJEXIST, *OBJMGT	*EXECUTE
RTVCLDSRC	Locale object	*USE	*EXECUTE
	To-file	See General Rules on page 311	See General Rules on page 311
RUNSQLSTM (SQL/400 licensed program) [1]	Source file	*OBJOPR, *READ	*EXECUTE
STRCBLDBG	Program	*CHANGE	*EXECUTE
STRREXPRC	Source file	*USE	*EXECUTE
	Exit program	*USE	*EXECUTE
STRSQL (DB2 Query Manager and SQL Development for OS/400 licensed program) [1]	Sort sequence table	*USE	*EXECUTE
	Printer device description	*USE	*EXECUTE
	Printer output queue	*USE	*EXECUTE
	Printer file	*USE	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library

[1] The DB2 Universal Database for iSeries topic in the Information Center contains more information about security requirements for structured query language (SQL) statements. See "Prerequisite and related information" on page xvi for details.

Library Commands

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	For Library Being Acted On
ADDLIBLE	Library		*USE
CHGCURLIB	New current library		*USE
CHGLIB [8]	Library		*OBJMGT
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	*OBJEXIST	*USE
	Object types *DTADCT[14], *JRN[14],*JRNRCV[14], *MSGQ, *SBSD	See the authority required by the DLTxxx command for the object type	
CPYLIB [4]	From-Library		*USE
	To-library, if it exists		*USE, *ADD
	CHKOBJ, CRTDUPOBJ commands	*USE	
	CRTLIB command, if the target library is being created	*USE	
	Object being copied	The authority that is required when you use the CRTDUPOBJ command to copy the object type.	
CRTLIB [9]	Library		
DLTLIB [3]	Every object being deleted from library	*OBJEXIST	*USE, *OBJEXIST
	Object types *DTADCT[14], *JRN[14],*JRNRCV[14], *MSGQ, *SBSD	See the authority required by the DLTxxx command for the object type	
DSPLIB	Library		*READ
	Objects in the library [5]	Any authority other than *EXCLUDE	
DSPLIBD	Library		Some authority other than *EXCLUDE
EDTLIBL	Library to add to list		*USE

Library Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library Being Acted On
RCLLIB	Library		*USE, *OBJEXIST
RSTLIB [7] (Q)	Media definition	*USE	*EXECUTE
	Library, if it does not exist		
	Message queues being restored to library where they already exist	*OBJOPR, *OBJEXIST [7]	*EXECUTE. *READ, *ADD
	Programs that adopt authority	Owner or *ALLOBJ and *SECADM	*EXECUTE
	Library saved if VOL(*SAVVOL) is specified		*USE [6]
	Every object being restored over in the library	*OBJEXIST [3]	*EXECUTE, *READ, *ADD
	User profile owning objects being created	*ADD [6]	
	Tape unit, diskette unit, optical unit	*USE	*EXECUTE
RSTLIB [7] (Q) (continued)	Tape (QSYSTAP) or diskette (QSYSDKT) file	*USE [6]	*EXECUTE
	QSYS/QPSRLDSP print file, if OUTPUT(*PRINT) specified	*USE	*EXECUTE
	Save file	*USE	*EXECUTE
	Optical File (OPTFILE)[12]	*R	N/A
	Path prefix of optical file (OPTFILE)[12]	*X	N/A
	Optical volume[11]	*USE	
RSTS36LIBM	From-file	*USE	*EXECUTE
	To-file	*CHANGE	*EXECUTE
	To-library	*CHANGE	*EXECUTE
	Device file or device description	*USE	*EXECUTE
RTVLIBD	Library		Some authority other than *EXCLUDE

Command	Referenced Object	Authority Needed	
		For Object	For Library Being Acted On
SAVLIB	Every object in the library	*OBJEXIST [6]	*READ, *EXECUTE
	Media definition	*USE	*EXECUTE
	Save file, if empty	*USE, *ADD	*EXECUTE
	Save file, if records exist in it	*USE, *ADD, *OBJMGT	*EXECUTE
	Save active message queue	*OBJOPR, *ADD	*EXECUTE
	Tape unit, diskette unit, optical unit	*USE	*EXECUTE
	Output file, if specified	See General Rules on page 311	See General Rules on page 311
	QSYS/QASAVOBJ field reference file, if output file is specified and does not exist	*USE [6]	*EXECUTE
	QSYS/QPSAVOBJ print file	*USE [6]	*EXECUTE
SAVLIB (continued)	Optical File[12]	*RW	N/A
	Parent Directory of optical file (OPTFILE)[12]	*WX	N/A
	Path Prefix of optical file (OPTFILE)[12]	*X	N/A
	Root Directory (/) of Optical Volume[12, 13]	*RWX	N/A
	Optical volume[11]	*CHANGE	
SAVS36LIBM	Save to a physical file	*OBJOPR, *OBJMGT	*EXECUTE
	Either QSYSDKT for diskette or QSYSTAP for tape, and all commands need authority to the device	*OBJOPR	*EXECUTE
	Save to a physical file if MBROPT(*ADD) is specified	*ADD	*READ, *ADD
	Save to a physical file if MBROPT(*REPLACE) is specified	*ADD, *DLT	*EXECUTE
	From-library		*USE
WRKLIB [10.]	Library		*USE

Library Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library Being Acted On
[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.			
[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] If you do not have authority to an object in the library, the text for the object says *NOT AUTHORIZED.			
[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. *OBJMGT is **not** required if you change only the CRTOBJAUD value. *OBJMGT **is** required if you change the CRTOBJAUD value and other values.			
[9] You must have *AUDIT special authority to specify a CRTOBJAUD value other than *SYSVAL.			
[10] You must have the authority required by th operation to use an individual operation.			
[11] Optical volumes are not actual system objects. The link between the optical volume and the authorization list used to secure the volume is maintained by the optical support function.			
[12] This authority check is only made when the Optical media format is Universal Disk Format.			
[13] This authority check is only made when you are clearing the optical volume.			
[14] This object is allowed on independent ASP.			

License Key Commands

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	For Library
ADDLICKEY (Q)	Output file	*USE	*EXECUTE
DSPLICKEY (Q)	Output file	See General Rules on page 311	See General Rules on page 311
RMVLICKEY (Q)	Output file	*CHANGE	*EXECUTE

Licensed Program Commands

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	For Library
CHGLICINF (Q)	WRKLICINF command	*USE	*EXECUTE
DLTLICPGM [1,2] (Q)			
DSPTM			
INZSYS (Q)			
RSTLICPGM [1,2] (Q)			
SAVLICPGM [1,2] (Q)			
WRKLICINF (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 Description Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGLINASC [2]	Line description	*CHANGE, *OBJMGT	*EXECUTE
	Controller description (SWTCTLLST)	*USE	*EXECUTE
CHGLINBSC [2]	Line description	*CHANGE, *OBJMGT	*EXECUTE
	Controller description (SWTCTLLST)	*USE	*EXECUTE
CHGLINDDI [2]	Line description	*CHANGE, *OBJMGT	*EXECUTE
CHGLINETH [2]	Line description	*CHANGE, *OBJMGT	*EXECUTE
CHGLINFAX [2]	Line description	*CHANGE, *OBJMGT	*EXECUTE
CHGLINFR [2]	Line description	*CHANGE, *OBJMGT	*EXECUTE
CHGLINIDLC [2]	Connection list (CNNLSTIN)	*USE	*EXECUTE
	Network interface description (SWTNWILST)	*USE	*EXECUTE
	Controller description (CTL)	*USE	*EXECUTE
	Line description	*CHANGE, *OBJMGT	*EXECUTE
CHGLINNET [2]	Line description	*CHANGE, *OBJMGT	*EXECUTE
CHGLINSDLC [2]	Line description	*CHANGE, *OBJMGT	*EXECUTE
CHGLINTDLC [2]	Line description	*CHANGE, *OBJMGT	*EXECUTE
CHGLINTRN [2]	Line description	*CHANGE, *OBJMGT	*EXECUTE
CHGLINX25 [2]	Line description	*CHANGE, *OBJMGT	*EXECUTE
	Controller description (SWTCTLLST)	*USE	*EXECUTE
	Connection list (CNNLSTIN or CNNLSTOUT)	*USE	*EXECUTE
	Network interface description (SWTNWILST)	*USE	*EXECUTE

Line Description Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGLINWLS [2]	Line description	*CHANGE, *OBJMGT	*EXECUTE
	Program (INZPGM)	*USE	*EXECUTE
CRTLINASC [2]	Controller description (CTL and SWTCTLLST)	*USE	*EXECUTE
	Line description		*READ, *ADD
CRTLINBSC [2]	Controller description (SWTCTLLST and CTL)	*USE	*EXECUTE
	Line description		*READ, *ADD
CRTLINDDI [2]	Line description		*READ, *ADD
	Network interface description (NWI)	*USE	*EXECUTE
	Controller description (NETCTL)	*USE	*EXECUTE
CRTLINETH [2]	Controller description (NETCTL)	*USE	*EXECUTE
	Line description		*READ, *ADD
	Network interface description (NWI)	*USE	*EXECUTE
	Network server description (NWS)	*USE	*EXECUTE
CRTLINFAX [2]	Line description		*READ, *ADD
	Controller description	*USE	*EXECUTE
CRTLINFR [2]	Line description		*READ, *ADD
	Network interface description (NWI)	*USE	*EXECUTE
	Controller description (NETCTL)	*USE	*EXECUTE
CRTLINIDLC [2]	Connection list (CNNLSTIN)	*USE	*EXECUTE
	Network interface description (NWI or SWTNWILST)	*USE	*EXECUTE
	Controller description (CTL)	*USE	*EXECUTE
	Line description		*READ, *ADD
CRTLINNET [2]	Network interface description (NWI)	*USE	*EXECUTE
	Controller description (CTL)	*USE	*EXECUTE
	Line description		*READ, *ADD
CRTLINSDLC [2]	Controller description (CTL)	*USE	*EXECUTE
	Line description		*READ, *ADD
CRTLINTDLC [2]	Controller description (WSC and CTL)	*USE	*EXECUTE
	Line description		*READ, *ADD
CRTLINTRN [2]	Controller description (NETCTL)	*USE	*EXECUTE
	Line description		*READ, *ADD
	Network interface description (NWI)	*USE	*EXECUTE
	Network server description (NWS)	*USE	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTLINX25 [2]	Controller description (SWTCTLLST)	*USE	*EXECUTE
	Permanent virtual circuit (PVC) controller description (LGLCHLE)	*USE	*EXECUTE
	Line description		*READ, *ADD
	Connection list (CNNLSTIN or CNNLSTOUT)	*USE	*EXECUTE
	Network interface description (NWI or SWTNWILST)	*USE	*EXECUTE
CRTLINWLS [2]	Line description		*READ, *ADD
	Controller description (NETCTL)	*USE	*EXECUTE
	Program (INZPGM)	*USE	*EXECUTE
DLTLIND	Line description	*OBJEXIST	*EXECUTE
DSPLIND	Line description	*USE	*EXECUTE
ENDLINRCY	Line description	*OBJOPR	*EXECUTE
PRTCMNSEC [2, 3]			
RSMLINRCY	Line description	*OBJOPR	*EXECUTE
WRKLIND [1]	Line description	*OBJOPR	*EXECUTE

[1] To use individual operations, you must have the authority required by the individual operation.

[2] To use this command, you must have *IOSYSCFG special authority.

[3] To use this command, you must have *ALLOBJ special authority.

Local Area Network (LAN) Commands

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.

These commands do not require any object authorities:			
ADDLANADPI	DSPLANADPP	RMVLANADPT (Q)	WRKLANADPT
CHGLANADPI	DSPLANSTS	RMVLANADPI	

Locale Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTLOCALE	Source file	*USE	*USE, *ADD
DLTLOCALE	Locale	*OBJEXIST	*USE

Mail Server Framework Commands

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.

Mail Server Framework Commands

These commands do not require any object authorities:	
ENDMSF (Q)	STRMSF (Q)

Media Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
ADDTAPCTG	Tape Library description	*USE	*EXECUTE
CFGDEVMLB[1] (Q)	Tape Library description	*USE	*EXECUTE
CHGDEVMLB (Q)	Tape Library description	*USE	*EXECUTE
CHGJOBMLBA	Tape Library description	*CHANGE	*EXECUTE
CHGTAPCTG	Tape Library description	*USE	*EXECUTE
CHKDKT	Diskette device description	*USE	*EXECUTE
CHKTAP	Tape device description	*USE	*EXECUTE
CLRDKT	Diskette device description	*USE	*EXECUTE
CRTTAPCGY	Tape Library description	*USE	*EXECUTE
DLTDKTLBL	Diskette device description	*USE	*EXECUTE
DLTMEDDFN	Media definition	*OBJEXIST	*EXECUTE
DLTTAPCGY	Tape Library description	*USE	*EXECUTE
DMPTAP	Tape device description	*USE	*EXECUTE
DSPDKT	Diskette device description	*USE	*EXECUTE
DSPTAP	Tape device description	*USE	*EXECUTE
DSPTAPCGY	Tape Library description	*USE	*EXECUTE
DSPTAPCTG	Tape Library description	*USE	*EXECUTE
DSPTAPSTS	Tape Library description	*USE	*EXECUTE
DUPDKT	Diskette device description	*USE	*EXECUTE
DUPTAP	Tape device description	*USE	*EXECUTE
INZDKT	Diskette device description	*USE	*EXECUTE
INZTAP	Tape device description	*USE	*EXECUTE
RMVTAPCTG	Tape Library description	*USE	*EXECUTE
RNMDKT	Diskette device description	*USE	*EXECUTE
SETTAPCGY	Tape Library description	*USE	*EXECUTE
WRKMLBRSCQ [3]	Tape Library description	*USE	*EXECUTE
WRKMLBSTS [2] (Q)	Tape Library description	*USE	*EXECUTE
WRKTAPCTG	Tape Library description	*USE	*EXECUTE

[1] To use this command, you must have *IOSYSCFG special authority.

[2] To use individual operation, you must have the authority required by the operation.

[2] To change the session media library attributes, you must have *CHANGE authority to the Tape Library description.

Menu and Panel Group Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGMNU	Menu	*CHANGE	*USE
CRTMNU	Source file	*USE	*EXECUTE
	Menu: REPLACE(*NO)		*READ, *ADD
	Menu: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
CRTPNLGRP	Panel group: Replace(*NO)		*READ, *ADD
	Panel group: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	Source file	*USE	*EXECUTE
	Include file	*USE	*EXECUTE
CRTS36MNU	Menu: REPLACE(*NO)		*READ, *ADD
	Menu: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	Source file	*USE	*EXECUTE
	Message files named in source	*OBJOPR, *OBJEXIST	*EXECUTE
	To-file source file when TOMBR is not *NONE	*OBJOPR, *OBJMGT, *OBJEXIST, *ADD	*READ, *ADD
	Menu display file when REPLACE(*YES) is specified	*OBJOPR, *OBJEXIST	*EXECUTE
	Command text message file	*OBJOPR, *OBJEXIST	*EXECUTE
	Create Message File (CRTMSGF) command	*OBJOPR	*EXECUTE
	Add Message Description (ADDMSGD) command	*OBJOPR	*EXECUTE
	Create Display File (CRTDSPF) command	*OBJOPR	*EXECUTE
DLTMNU	Menu	*OBJOPR, *OBJEXIST	*EXECUTE
DLTPNLGRP	Panel group	*OBJEXIST	*EXECUTE
DSPMNUA	Menu	*USE	*USE
GO	Menu	*USE	*USE
	Display file and message files with *DSPF specified	*USE	*EXECUTE
	Current and Product libraries	*USE	
	Program with *PGM specified	*USE	*EXECUTE
WRKMNU [1]	Menu	Any	*USE
WRKPNLGRP [1]	Panel group	Any	*EXECUTE

[1] To use an individual operation, you must have the authority required by the operation .

Message Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
DSPMSG	Message queue	*USE	*USE
	Message queue that receives the reply to an inquiry message	*USE, *ADD	*USE
	Remove messages from message queue	*USE, *DLT	*USE
RCVMSG	Message queue	*USE	*EXECUTE
	Remove messages from queue	*USE, *DLT	*EXECUTE
RMVMSG	Message queue	*OBJOPR, *DLT	*EXECUTE
RTVMSG	Message file	*USE	*EXECUTE
SNDBRKMSG	Message queue that receives the reply to inquiry messages	*OBJOPR, *ADD	*EXECUTE
SNDMSG	Message queue	*OBOPR, *ADD	*EXECUTE
	Message queue that receives the reply to inquiry message	*OBJOPR, *ADD	*EXECUTE
SNDPGMMSG	Message queue	*OBJOPR, *ADD	*EXECUTE
	Message file, when sending predefined message	*USE	*EXECUTE
	Message queue that receives the reply to inquiry message	*OBJOPR, *ADD	*EXECUTE
SNDRPY	Message queue	*USE, *ADD	*EXECUTE
	Remove messages from queue	*USE, *ADD, *DLT	*EXECUTE
SNDUSRMSG	Message queue	*OBJOPR, *ADD	*EXECUTE
	Message file, when sending predefined message	*USE	*EXECUTE
WRKMSG	Message queue	*USE	*USE
	Message queue that receives the reply to inquiry message	*USE, *ADD	*USE
	Remove messages from message queue	*USE, *DLT	*USE

Message Description Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
ADDMSGD	Message file	*USE, *ADD	*EXECUTE
CHGMSGD	Message file	*USE, *UPD	*EXECUTE
DSPMSGD	Message file	*USE	*EXECUTE
RMVMSGD	Message file	*OBJOPR, *DLT	*EXECUTE
WRKMSGD [1]	Message file	*USE	*EXECUTE

[1] To use individual operations, you must have the authority required by the individual operation.

Message File Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGMSGF	Message file	*USE, *DLT	*EXECUTE
CRTMSGF	Message file		*READ, *ADD
DLTMSGF	Message file	*OBJEXIST	*EXECUTE
DSPMSGF	Message file	*USE	*EXECUTE
MRGMSGF	From-message file	*USE	*EXECUTE
	To-message file	*USE, *ADD, *DLT	*EXECUTE
	Replace-message file	*USE, *ADD	*EXECUTE
WRKMSGF [1.]	Message file	Any authority	*USE
[1] To use an individual operation, you must have the authority required by the operation .			

Message Queue Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGMSGQ	Message queue	*USE, *DLT	*EXECUTE
CLRMSGQ	Message queue	*OBJOPR, *DLT	*EXECUTE
CRTMSGQ	Message queue		*READ, *ADD
DLTMSGQ	Message queue	*OBJEXIST, *USE, *DLT	*EXECUTE
DSPLOG			*EXECUTE
WRKMSGQ [1]	Message queue	Any authority	*USE
[1] To use an individual operation, you must have the authority required by the operation .			

Migration Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
RCVMGRDTA	File	*ALL	*READ, *ADD
	Device	*CHANGE	*EXECUTE
SNDMGRDTA	File	*ALL	*READ, *ADD
	Device	*CHANGE	*EXECUTE
The following commands do not require any object authorities. They are shipped with public authority *EXCLUDE. You must have *ALLOBJ special authority to use these commands.			

Migration Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
ANZS34OCL	CVTS36JOB	MGRS36DSPF	MIGRATE
ANZS36OCL	CVTS36QRY	MGRS36ITM	QMUS36
CHGS34LIBM	CVTS38JOB	MGRS36LIB	RESMGRNAM
CHKS36SRCA	GENS36RPT	MGRS36MNU	RSTS38AUT
CVTBASSTR	GENS38RPT	MGRS36MSGF	STRS36MGR
CVTBASUNF	MGRS36	MGRS36QRY [1]	STRS38MGR
CVTBGUDTA	MGRS36APF [1]	MGRS36RPG	
CVTS36CFG	MGRS36CBL	MGRS36SEC	
CVTS36FCT	MGRS36DFU [1]	MGRS38OBJ	

[1] You must have *ALLOBJ special authority and have OS/400 option 4 installed.

Mode Description Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGMODD [2]	Mode description	*CHANGE, *OBJMGT	*EXECUTE
CRTMODD [2]	Mode description		*READ, *ADD
CHGSSNMAX	Device description	*OBJOPR	*EXECUTE
DLTMODD	Mode description	*OBJEXIST	*EXECUTE
DSPMODD	Mode description	*USE	*EXECUTE
DSPMODSTS	Device	*OBJOPR	*EXECUTE
	Mode description	*OBJOPR	*EXECUTE
ENDMOD	Device description	*OBJOPR	*EXECUTE
STRMOD	Device description	*OBJOPR	*EXECUTE
WRKMODD [1]	Mode description	*OBJOPR	*EXECUTE

[1] To use individual operations, you must have the authority required by the individual operation.

[2] To use this command, you must have *IOSYSCFG special authority.

Module Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGMOD	Module	*OBJMGT, *USE	*USE
	Module, if OPTIMIZE specified	*OBJMGT, *USE	*USE, *ADD, *DLT
	Module, if FRCCRT(*YES) specified	*OBJMGT, *USE	*USE, *ADD, *DLT
	Module, if ENBPRFCOL specified	*OBJMGT, *USE	*USE, *ADD, *DELETE
DLTMOD	Module	*OBJEXIST	*EXECUTE
DSPMOD	Module	*USE	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
RTVBNDSRC [1]	Module	*USE	*EXECUTE
	*SRVPGMs and modules specified with *SRVPGMs	*USE	*EXECUTE
	Database source file if file and member exists and MBROPT(*REPLACE) is specified.	*OBJOPR, *OBJMGT, *ADD, *DLT	*EXECUTE
	Database source file if file and member exists and MBROPT(*ADD) is specified	*OBJOPR, *ADD	*EXECUTE
	Database source file if file exists and member needs to be created.	*OBJOPR, *OBJMGT, *ADD	*EXECUTE, *READ, *ADD
	Database source file if file and member needs to be created.		*EXECUTE, *READ, *ADD
	CRTSCRPF command if file does not exist		*EXECUTE
	ADDPFM command if member does not exist		*EXECUTE
	RGZPFM command to reorganize source file member	*OBJMGT	*EXECUTE
WRKMOD [2]	Module	Any authority	*USE

[1] You need *USE authority to the:

- CRTSRCPF command if the file does not exist.
- ADDPFM command if the member does not exist.
- RGZPFM command so the source file member is reorganized. Either *CHANGE and *OBJALTER authorities or *OBJMGT authority is required to reorganize the source file member. The RTVBNDSRC command function then completes with the source file member reorganized with sequence numbers of zero.

[2] To use an individual operation, you must have the authority required by the operation

NetBIOS Description Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGNTBD [2]	NetBIOS description	*CHANGE, *OBJMGT	*EXECUTE
CRTNTBD [2]	NetBIOS description		*EXECUTE
DLTNTBD	NetBIOS description	*OBJEXIST	*EXECUTE
DSPNTBD	NetBIOS description	*USE	*EXECUTE
WKRNTBD[1]	NetBIOS description	*OBJOPR	*EXECUTE

[1] To use an individual operation, you must have the authority required by the operation

[2] To use this command, you must have *IOSYSCFG special authority.

Network Commands

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.

Network Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
ADDNETJOBE (Q)	User profile in the network job entry	*USE	
APING	Device description	*CHANGE	
AREXEC	Device description	*CHANGE	
CHGNETA (Q) [4]			
CHGNETJOBE (Q)	User profile in the network job entry	*USE	
DLTNETF [2]	Output file	See General Rules on page 311	See General Rules on page 311
DSPNETA			
RCVNETF [2]	To-file member does not exist, MBROPT(*ADD) specified	*OBJMGT, *USE	*EXECUTE, *ADD
	To-file member does not exist, MBROPT(*REPLACE) specified	*OBJMGT, *CHANGE	*EXECUTE, *ADD
	To-file member exists, MBROPT(*ADD) specified	*USE	*EXECUTE
	To-file member exists, MBROPT(*REPLACE) specified	*OBJMGT, *CHANGE	*EXECUTE
RMVNETJOBE (Q)	User profile in the network job entry	*USE	
RTVNETA			
RUNRMTCMD	Device description	*CHANGE	
SNDNETF	Physical file or save file	*USE	*EXECUTE
SNDNETMSG to a local user	Message queue	*OBJOPR, *ADD	*EXECUTE
VFYAPPCCNN	Device description	*CHANGE	
WRKNETF [2,3]			
WRKNETJOBE [3]	QUSRSYS/QANFNJE	*USE	*EXECUTE

[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] To change some network attributes, you must have *ALLOBJ and *IOSYSCFG special authorities.

Network File System Commands

Command	Referenced Object	Object Type	File System	Authority Needed for Object
ADDMFS [1,2,3]	/dev/QASPxx	*DIR	"root"	*RWX
	/dev/QASPxx/yyy	*BLKSF	"root"	*R
	dir_to_be_ mounted_over	*DIR	"root"	*RWX

Command	Referenced Object	Object Type	File System	Authority Needed for Object
CHGNFSEXP [1,4,5]	some_dirs	*DIR	"root"	*RX
	/etc	*DIR	"root"	*RWX
	/etc/exports	*STMF	"root"	*RWX
	/etc/netgroup	*STMF	"root"	*RWX
DSPMFSINF [1,4]	some_dirs	*DIR	"root"	*RX
ENDNFSSVR [1]	none			
EXPORTFS [1,4,5]	some_dirs	*DIR	"root"	*RX
	/etc	*DIR	"root"	*RWX
	/etc/exports	*STMF	"root"	*RWX
	/etc/netgroup	*STMF	"root"	*RWX
MOUNT [1,2,3]	/dev/QASPxx	*DIR	"root"	*RWX
	/dev/QASPxx/yyy	*BLKSF[8]	"root"	*R
	dir_to_be_ mounted_over	*DIR	"root"	*RWX
RLSIFSLCK [1]	some_dirs	*DIR	"root"	*RX
	some_stmf	*STMF	"root"	*RWX
RMVMFS [1]	some_dirs	*DIR	"root"	*RX
STATFS [1,4]	some_dirs	*DIR	"root"	*RX
STRNFSSVR [1]	none			
UNMOUNT [1]	some_dirs	*DIR	"root"	*RX

1 To use this command, you must have *IOSYSCFG special authority.

2 QASPxx is either 01 (system asp) or 02-16 based on which user asp is needed. This is the directory that contains the *BLKSF that is being mounted.

3 The directory that is mounted over (dir_to_be_mounted_over) is any IFS directory that can be mounted over.

4 You must provide a path to some object. You must have *RX authority for all directories in that path.

5 You must have *RX authority to the /etc/exports stream file and the directories in the /etc/exports path.

6 You must provide a path to some *STMF. You must have *RX authority for all directories in that path.

7 You must have update (*RWX) authority to the stream file for which you are releasing locks.

8 This object is allowed on independent disk pools.

Network Interface Description Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGNWIFR [2]	Network interface description	*CHANGE, *OBJMGT	*EXECUTE
CHGNWIISDN [2]	Network interface description	*CHANGE, *OBJMGT	*EXECUTE
	Line description (CHLENTRY)	*USE	*EXECUTE
CRTNWIFR [2]	Network interface description		*READ, *ADD
	Line description (DLCI)	*USE	*EXECUTE

Network Interface Description Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTNWIISDN [2]	Network interface description	*USE	*EXECUTE
	Line description (CHLENTRY)	*USE	*EXECUTE
DLTNWID	Network interface description	*OBJEXIST	*EXECUTE
DSPNWID	Network interface description	*USE	*EXECUTE
WRKNWID [1]	Network interface description	*OBJOPR	*EXECUTE

[1] To use the individual operations, you must have the authority required by the individual operation.

[2] To use this command, you must have *IOSYSCFG special authority.

Network Server Commands

Command	Referenced Object	Object Type	File System	Authority Needed for Object
ADDNWSSTGL [2]	Path (/QFPNWSSTG)	*DIR	"root"	*X
	Parent directory (name of the storage space)	*DIR	"root"	*WX
	Files that make up the storage space	*FILE	"root"	*RW
	Network server description	*NWSD	QSYS.LIB	*CHANGE, *OBJMGT
CHGNWSUSRA [4]	User Profile	*USRPRF		*OBJMGT, *USE
CRTNWSSTG [2]	Path (root and /QFPNWSSTG)	*DIR	"root"	*WX
DLTNWSSTG [2]	Path (/QFPNWSSTG)	*DIR	"root"	*WX
	Parent directory (name of the storage space)	*DIR	"root"	*RWX, *OBJEXIST
	Files that make up the storage space	*FILE	"root"	*OBJEXIST
DSPNWSSTG	Path to the storage space	*DIR	"root"	*X
	Files that make up the storage space	*FILE	"root"	*R
RMVNWSSTGL [2]	Path (/QFPNWSSTG)	*DIR	"root"	*X
	Parent directory (name of the storage space)	*DIR	"root"	*WX
	Files that make up the storage space	*FILE	"root"	*RW
	Network server description	*NWSD	QSYS.LIB	*CHANGE, *OBJMGT
WRKNWSSTG	Path to the storage space	*DIR	"root"	*X
	Files that make up the storage space	*FILE	"root"	*R

These commands do not require any object authorities:

ADDRMTSVR	DSPNWSALS	SNDNWSMSG
CHGNWSA [4](Q)	DSPNWSSSN	WRKNWSALS
CHGNWSALS	DSPNWSSTC	WRKNWSENR
CRTNWSALS	DSPNWSUSR	WRKNWSSSN
DLTNWSALS	DSPNWSUSRA	WRKNWSSTS
DSPNWSA	SBMNWSCMD (Q) [3]	

Command	Referenced Object	Object Type	File System	Authority Needed for Object

[1] Adopted authority is not used for Network Server commands.

[2] To use this command, you must have *IOSYSCFG special authority.

[3] To use this command, you must have *JOBCTL special authority.

[4] You must have *SECADM special authority to specify a value other than *NONE for the NDSTREELST and the NTW3SVRLST paramaters.

Network Server Description Commands

Command	Referenced Object	Authority Needed	
		For Object	For QSYS Library
CHGNWSD [2]	Network server description	*CHANGE, *OBJMGT	*EXECUTE
	NetBIOS description (NTB)	*USE	*EXECUTE
CRTNWSD [2]	NetBIOS description (NTB)	*USE	*EXECUTE
	Line description (PORTS)	*USE	*EXECUTE
DLTNWSD	Network server description	*OBJEXIST	*EXECUTE
DSPNWSD	Network server description	*USE	*EXECUTE
WRKNWSD[1]	Network server description	*OBJOPR	*EXECUTE

[1] To use an individual operation, you must have the authority required by the operation

[2] To use this command, you must have *IOSYSCFG special authority.

Node List Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
ADDNODLE	Node list	*OBJOPR, *ADD	*EXECUTE
CRTNODL	Node list		*READ, *ADD
DLTNODL	Node list	*OBJEXIST	*EXECUTE
RMVNODLE	Node list	*OBJOPR, *READ, *DLT	*EXECUTE
WRKNODL [1]	Node list	*USE	*USE
WRKNODLE	Node list	*USE	*EXECUTE

[1] To use the individual operations, you must have the authority required by the individual operation.

Office Services Commands

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.

These commands do not require object authorities.

Office Services Commands

ADDACC (Q)	GRTACCAUT [1,2,3] (Q)	RVKUSRPMN [1,2]
DSPACC	GRTUSRPMN [1,2]	WRKDOCLIB [4]
DSPACCAUT	RMVACC [1] (Q)	WRKDOCPRTQ [5]
DSPUSRPMN	RVKACCAUT [1]	

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 Commands

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.

		Authority Needed	
Command	**Referenced Object**	**For Object**	**For Library**
CVTEDU			
STREDU			

Operational Assistant Commands

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.

		Authority Needed	
Command	**Referenced Object**	**For Object**	**For Library**
CHGBCKUP [1]	QUSRSYS/QEZBACKUPL *USRIDX	*CHANGE	*EXECUTE
CHGCLNUP[2]			
CHGPWRSCD [3]	PWRDWNSYS *CMD	*USE	*EXECUTE
CHGPWRSCDE [3]	PWRDWNSYS *CMD	*USE	*EXECUTE
DSPBCKSTS	QUSRSYS/QEZBACKUPL *USRIDX	*USE	*EXECUTE
DSPBCKUP	QUSRSYS/QEZBACKUPL *USRIDX	*USE	*EXECUTE
DSPBCKUPL	QUSRSYS/QEZBACKUPL *USRIDX	*USE	*EXECUTE
	QUSRSYS/QEZBACKUPF *USRIDX	*USE	*EXECUTE
DSPPWRSCD			*EXECUTE
EDTBCKUPL [1]	QUSRSYS/QEZBACKUPL *USRIDX	*CHANGE	*EXECUTE
	QUSRSYS/QEZBACKUPF *USRIDX	*CHANGE	*EXECUTE
ENDCLNUP [4]	ENDJOB *CMD	*USE	*EXECUTE
PRTDSKINF (Q)	QUSRSYS/QAEZDISK *FILE, member QCURRENT	*USE	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
RTVBCKUP	QUSRSYS/QEZBACKUPL *USRIDX	*USE	*EXECUTE
RTVCLNUP			
RTVDSKINF (Q) [5]			
RTVPWRSCDE	DSPPWRSCD command	*USE	*EXECUTE
RUNBCKUP [1]	QUSRSYS/QEZBACKUPL *USRIDX	*USE	*EXECUTE
	QUSRSYS/QEZBACKUPF *USRIDX	*USE	*EXECUTE
	Commands: SAVLIB, SAVCHGOBJ, SAVDLO, SAVSECDTA, SAVCFG, SAVCAL, SAV	*USE	*EXECUTE
STRCLNUP[4]	QPGMR User profile	*USE	
	Job queue	*USE	*EXECUTE

[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.

Optical Commands

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.

Table 139.

Command	Referenced Object	Authority Needed		
		Object	Library	Optical Volume [1]
ADDOPTCTG (Q)	Optical Device	*USE	*EXECUTE	
ADDOPTSVR (Q)	Server CSI	*USE	*EXECUTE	
CHGDEVOPT[4]	Optical Device	*CHANGE, *OBJMGT	*EXECUTE	
CHGOPTA (Q)				
CHGOPTVOL	Root directory (/) of volume when changing the Text Description[5]	*W	N/A	N/A
	Optical Device	*USE	N/A	N/A
	Server CSI	*USE	N/A	N/A

Table 139. (continued)

Command	Referenced Object	Authority Needed		
		Object	Library	Optical Volume [1]
CPYOPT	Optical Device	*X	N/A	N/A
	Each preceding dir in path of source file	*X	N/A	N/A
	Each preceding dir in path of target file	*X	N/A	N/A
	Source file (*DSTMF)[5]	*R	N/A	N/A
	Parent dir of target file	*WX	N/A	N/A
	Parent of parent dir if creating dir	*WX	N/A	N/A
CPYOPT (continued)	Target file if replaced due to SLTFILE(*ALL)	*W	N/A	N/A
	Target file if replaced due to SLTFILE(*CHANGED)	*RW	N/A	N/A
	Each dir in path that precedes source dir	*X	N/A	N/A
	Each dir in path that precedes target dir	*X	N/A	N/A
CPYOPT (continued)	Dir being copied[5]	*R	N/A	N/A
	Dir being copied if it contains entries	*RX	N/A	N/A
	Parent of target dir	*WX	N/A	N/A
	Target dir if replaced due to SLTFILE(*ALL)	*W	N/A	N/A
	Target dir if replaced due to SLTFILE(*CHANGED)	*RW	N/A	N/A
	Target dir if entries are to be created	*WX	N/A	N/A
COPYOPT (continued)	Source files	*R	N/A	N/A
	Target file if replaced due to SLTFILE(*ALL)	*W	N/A	N/A
	Target file if replaced due to SLTFILE(*CHANGED)	*RW	N/A	N/A
CRTDEVOPT[4]	Optical Device		*EXECUTE	*ALL - Target Volume
CVTOPTBKU	Optical Device	*USE	*EXECUTE	*ALL

Table 139. (continued)

Command	Referenced Object	Authority Needed		
		Object	Library	Optical Volume [1]
DSPOPT	Path Prefix when DATA (*SAVRST)[5]	*X	N/A	N/A
	File Prefix when (*SAVRST)[2]	*R	N/A	N/A
	Optical Device	*EXECUTE	*USE	
	Server CSI	*USE	*EXECUTE	
DSPOPTLCK				
DSPOPTSVR	Server CSI	*USE	*EXECUTE	
DUPOPT	Optical Device	*USE	*EXECUTE	*USE - Source Volume
				*ALL - Target Volume
INZOPT	Root directory (/) of volume	*RWX	N/A	N/A
	Optical Device	*USE	*EXECUTE	*ALL
RCLOPT (Q)	Optical Device	*USE	*EXECUTE	
RMVOPTCTG (Q)	Optical Device	*USE	*EXECUTE	
RMVOPTSVR (Q)	Server CSI	*USE	*EXECUTE	
WRKHLDOPTF[2]	Optical Device	*USE	*EXECUTE	*USE
	Server CSI	*USE	*EXECUTE	
WRKOPTDIR[2]	Optical Device	*USE	*EXECUTE	*USE
	Server CSI	*USE	*EXECUTE	
WRKOPTF[2]	Optical Device	*USE	*EXECUTE	*USE
	Server CSI	*USE	*EXECUTE	
WRKOPTVOL[2]	Optical Device	*USE	*EXECUTE	

[1] Optical volumes are not actual system objects. The link between the optical volume and the authorization list used to secure the volume is maintained by the optical support function.

[2] There are seven options that can be invoked from the optical utilities that are not commands themselves. These options and their required authorities to the optical volume are shown below.

Delete File: *CHANGE

Rename File: *CHANGE

Delete Directory: *CHANGE

Create Directory: *CHANGE

Rename Volume: *ALL

Release Held Optical File: *CHANGE

Save Held Optical File: *USE - Source Volume, *Change - Target Volume

[3] Authorization list management authority to the authorization list currently securing the optical volume is needed to change the authorization list used to secure the volume.

[4] To use this command, you must have *IOSYSCFG special authority.

[5] This authority check is only made when the Optical media format is Universal Disk Format (UDF).

Output Queue Commands

Command	Referenced Object	Authority Needed		Output Queue Parameters		Special Authority
		For Object	For Library	AUTCHK	OPRCTL	
CHGOUTQ [1]	Data queue	*READ	*EXECUTE			
	Output queue	*OBJMGT, *READ, *ADD, *DLT	*EXECUTE	*DTAAUT		
		Owner [2]	*EXECUTE	*OWNER		
			*EXECUTE		*YES	*JOBCTL
CLROUTQ [1]	Output queue	*READ, *ADD, *DLT	*EXECUTE	*DTAAUT		
		Owner [2]	*EXECUTE	*OWNER		
			*EXECUTE		*YES	*JOBCTL
CRTOUTQ	Data queue	*READ	*EXECUTE			
	Output queue		*READ, *ADD			
DLTOUTQ	Output queue	*OBJEXIST	*EXECUTE			
HLDOUTQ [1]	Output queue	*READ, *ADD, *DLT	*EXECUTE	*DTAAUT		
		Owner [2]	*EXECUTE	*OWNER		
			*EXECUTE		*YES	*JOBCTL
PRTQAUT [4]						
RLSOUTQ [1]	Output queue	*READ, *ADD, *DLT	*EXECUTE	*DTAAUT		
		Owner [2]	*EXECUTE	*OWNER		
			*EXECUTE		*YES	*JOBCTL
WRKOUTQ [1,3]	Output queue	*READ	*EXECUTE			
			*EXECUTE		*YES	*JOBCTL
WRKOUTQD [1,3]	Output queue	*READ	*EXECUTE			
			*EXECUTE		*YES	*JOBCTL

1 If you have *SPLCTL special authority, you do not need authority to the output queue. You do need *EXECUTE authority, however, to the library for the outqueue.

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 *EXECUTE authority.

4 You must have *ALLOBJ special authority to use this command.

Package Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTSQLPKG	Program	*OBJOPR, *READ	*EXECUTE
	SQL package: REPLACE(*NO)		*OBJOPR, *READ, *ADD, *EXECUTE
	SQL package: REPLACE(*YES)	*OBJOPR, *OBJMGT, *OBJEXIST, *READ	*OBJOPR, *READ, *ADD, *EXECUTE
DLTSQLPKG	Package	*OBJEXIST	*EXECUTE
PRTSQLINF	Package	*OBJOPR, *READ	*EXECUTE
	Program	*OBJOPR, *READ	*EXECUTE
	Service program	*OBJOPR, *READ	*EXECUTE
STRSQL			

Performance Commands

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 to others.

Command	Referenced Object	Authority Needed	
		For Object	For Library
ADDPEXDFN (Q) [5]	PGM Library		*EXECUTE
ADDPEXFTR (Q)[5]	PGMTRG Library		*EXECUTE
	PGMFTR Library		*EXECUTE
	JVAFTR Path	*X for directory	
	PATHFTR Path	*X for directory	
ANZACCGRP (Q) [4]	QPFR/QPTPAGA0 *PGM	*USE	*EXECUTE
	Model library		*EXECUTE, *ADD
	Job description	*USE	*EXECUTE
	QPFR/QCYRBCPP *PGM	*USE	*EXECUTE
	QPFR/QCYMBREX *PGM	*USE	*EXECUTE
ANZBESTMDL (Q)[4]	QPFR/QCYRBMN *PGM	*USE	*EXECUTE
	Application libraries that contain the database files to be analyzed		*EXECUTE
	Job description	*USE	*EXECUTE
ANZDBF (Q)[4]	QPFR/QCYRBMN *PGM	*USE	*EXECUTE
	Job description	*USE	*EXECUTE
ANZDBFKEY (Q)	QPFR/QPTANZKC *PGM	*USE	*EXECUTE
	Application libraries that contain the programs to be analyzed		*EXECUTE
	Job description	*USE	*EXECUTE
ANZPGM (Q)	QPFR/QPTANZPC *PGM	*USE	*EXECUTE
	Performance data[2]		*ADD, *READ

Performance Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
ANZPFRDTA (Q)[4]	QPFR/QACVPP *PGM	*USE	*EXECUTE
	Performance data[2]		*ADD, *READ
ANZPFRDT2 (Q)[4]	QPFR/QAVCPP *PGM	*USE	*EXECUTE
	QAPTAPGP *FILE	*CHANGE	*EXECUTE
	DLTFCNARA command (Q)	*USE	*EXECUTE
	QPFR/QPTAGRP *PGM	*USE	*EXECUTE
CHGFCNARA (Q)	QPFR/QPTAGRPD *PGM	*USE	*EXECUTE
	QAPGGPHF *FILE	*CHANGE	*EXECUTE
CHGGPHFMT (Q)	QPFR/QPGCRTFM *PGM	*USE	*EXECUTE
	QAPGPKGF *FILE	*CHANGE	*EXECUTE
	QAPGGPHF *FILE	*USE	*EXECUTE
CHGGPHPKG (Q)	QPFR/QPGCRTPK *PGM	*USE	*EXECUTE
	QAPMDMPT *FILE	*CHANGE	*EXECUTE
CHGJOBTYP (Q)	QPFR/QPTCHGJT *PGM	*USE	*EXECUTE
CHGPEXDFN (Q)[5]	PGM Library		*EXECUTE
CPYFCNARA (Q)[4]	QPFR/QPTAGRPR *PGM	*USE	*EXECUTE
	QAPGGPHF *FILE in ″From″ library	*USE	*EXECUTE
	″To″ library (if QAPGGPHF *FILE does not exist)		*EXECUTE, *ADD
	QAPGGPHF *FILE in ″To″ library (if adding a new graph format or replacing an existing one)	*CHANGE	*EXECUTE
CPYGPHFMT (Q)[4]	QPFR/QPGCPYGP *PGM	*USE	*EXECUTE
	QAPGPKGF *FILE in ″From″ library	*USE	*EXECUTE
	″To″ library (if QAPGPKGF *FILE does not exist)		*EXECUTE, *ADD
	QAPGPKGF *FILE in ″To″ library (if adding a new graph package or replacing an existing one)	*CHANGE	*EXECUTE
	QAPGGPHF *FILE in ″To″ library (if adding a new graph package or replacing an existing one)	*USE	*EXECUTE
CPYGPHPKG (Q)	QPFR/QPGCPYGP *PGM	*USE	*EXECUTE
	From library		*EXECUTE
	To library		*EXECUTE, *ADD
	Job description	*USE	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
CPYPFRDTA (Q)	QPFR/QITCPYCP *PGM	*USE	*EXECUTE
	Performance data (all QAPM* files)	*USE	*EXECUTE
	Model library		*EXECUTE, *ADD
	Job description	*USE	*EXECUTE
	QPFR/QCYCBMCP *PGM	*USE	*EXECUTE
	QPFR/QCYCBMDL *PGM	*USE	*EXECUTE
	QPFR/QCYOPDBS *PGM	*USE	*EXECUTE
	QPFR/QCYCLIDS *PGM	*USE	*EXECUTE
CRTBESTMDL (Q)	QPFR/QCYCAPT *PGM	*USE	*EXECUTE
	Library where the Functional Area is created		*EXECUTE, *ADD
	QAPTAPGP *FILE in target library (if adding a new functional area)	*CHANGE	*EXECUTE
CRTFCNARA (Q)	QPFR/QPTAGRP *PGM	*USE	*EXECUTE
	Library where the Graph Format is created		*EXECUTE, *ADD
	QAPGGPHF *FILE in target library (if adding a new graph format)	*CHANGE	*EXECUTE
CRTGPHFMT (Q)	QPFR/QPGCRTFM *PGM	*USE	*EXECUTE
	Library where the Graph Package is created		*EXECUTE, *ADD
	QAPGGPHF *FILE	*CHANGE	*EXECUTE
	QAPGPKGF *FILE in target library (if adding a new graph package)	*USE	*EXECUTE
CRTGPHPKG (Q)	QPFR/QPGCRTPK *PGM	*USE	*EXECUTE
	Library where the historical data is created		*ADD, *READ
	Job description	*USE	*EXECUTE
CRTHSTDTA (Q)	QPFR/QPGCRTHS *PGM	*USE	*EXECUTE
	To Library		*ADD, *READ
CRTPEXDTA (Q)[5]	*MGTCOL Library		*EXECUTE
	Data library[1]		*READ,*ADD[2]
CRTPFRDTA (Q)	From Library		*EXECUTE
	To Library		*ADD, *READ
	From Library		*USE
CVTPFRDTA (Q)	Job description	*USE	*EXECUTE
CVTPFRTHD (Q)	Performance data[2]		*ADD, *READ
	Model library		*EXECUTE, *ADD
	QPFR/QCYDBMDL *PGM	*USE	*EXECUTE
	QPFR/QCYCVTBD *CMD	*USE	*EXECUTE
DLTBESTMDL (Q)[4]	QPFR/QCYCBTOD *PGM	*USE	*EXECUTE
	QAPTAPGP *FILE in the functional area library	*CHANGE	*EXECUTE

Performance Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
DLTFCNARA (Q)[4]	QPFR/QPTAGRPD *PGM	*USE	*EXECUTE
	QAPGGPHF *FILE in the graph format library	*CHANGE	*EXECUTE
DLTGPHFMT (Q)[4]	QPFR/QPGDLTGP *PGM	*USE	*EXECUTE
	QAPGPKGF *FILE in the graph package library	*CHANGE	*EXECUTE
DLTGPHPKG (Q)[4]	QPFR/QPGDLTGP *PGM	*USE	*EXECUTE
	QAPGHSTD *FILE in the historical data library	*CHANGE	*EXECUTE
	QAPGHSTI *FILE in the historical data library	*CHANGE	*EXECUTE
	QAPGSUMD *FILE in the historical data library	*CHANGE	*EXECUTE
DLTHSTDTA (Q)[4]	QPFR/QPGDLTHS *PGM	*USE	*EXECUTE
DLTPEXDTA (Q)[5]	Data Library [1]		*EXECUTE, *DELETE [2]
DLTPFRDTA (Q)[4]	QPFR/QPTDLTCP *PGM	*USE	*EXECUTE
DMPTRC (Q)[5]	Library where the trace data will be stored		*EXECUTE, *ADD
	Output file (QAPTPAGD)	*CHANGE	*EXECUTE, *ADD
DSPACCGRP (Q)[4]	QPFR/QPTPAGD0 *PGM	*USE	*EXECUTE
	Format or package library		*EXECUTE
	Historical data library		*EXECUTE
	Output file library		*EXECUTE, *ADD
	Output queue	*USE	*EXECUTE
	Job description	*USE	*EXECUTE
DSPHSTGPH (Q)[4]	QPFR/QPGCTRL *PGM	*USE	*EXECUTE
	Historical data library		*EXECUTE
DSPPFRDTA (Q) [4]	QPFR/QAVCPP *PGM	*USE	*EXECUTE
	Format or package library		*EXECUTE
	Performance data[2]		*EXECUTE
	Output file library		*EXECUTE, *ADD
	Output queue	*USE	*EXECUTE
	Job description	*USE	*EXECUTE
DSPPFRGPH (Q)[4]	QPFR/QPGCTRL *PGM	*USE	*EXECUTE
	Output file library		*EXECUTE
	Job description	*USE	*EXECUTE
ENDJOBTRC (Q)[4]	QPFR/QPTTRCJ0 *PGM	*USE	*EXECUTE
ENDPEX (Q)[5]	Data Library [1]		*READ, *ADD [2]
PRTACTRPT (Q)[4]	QPFR/QITPRTAC *PGM	*USE	*EXECUTE
	Performance data[2]	*USE	*ADD, *READ
	Job description	*USE	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
PRTCPTRPT (Q)[4]	QPFR/QPTCPTRP *PGM	*USE	*EXECUTE
	Performance data[2]		*ADD, *READ
	Job description	*USE	*EXECUTE
PRTJOBRPT (Q)[4]	QPFR/QPTITVXC *PGM	*USE	*EXECUTE
	Performance data[2]		*ADD, *READ
	Job description	*USE	*EXECUTE
PRTJOBTRC (Q)[4]	QPFR/QPTTRCRP *PGM	*USE	*EXECUTE
	Job trace file (QAPTTRCJ) library		*EXECUTE
	Job description	*USE	*EXECUTE
PRTLCKRPT (Q)[4]	QPFR/QPTLCKQ *PGM	*USE	*EXECUTE
PRTPEXRPT[5]	Data Library[1]		*EXECUTE[2]
	Outfile	*USE	*EXECUTE,*ADD
	QPFR/QVPEPRTC *PGM	*USE	*EXECUTE
	QPFR/QVPESVGN *SRVPGM	*USE	*EXECUTE
	QPFR/QYPESVGN *SRVPGM	*USE	*EXECUTE
PRTPOLRPT (Q)[4]	QPFR/QPTITVXC *PGM	*USE	*EXECUTE
	Performance data[2]		*ADD, *READ
	Job description	*USE	*EXECUTE
PRTRSCRPT (Q)[4]	QPFR/QPTITVXC *PGM	*USE	*EXECUTE
	Performance data[2]		*ADD, *READ
	Job description	*USE	*EXECUTE
PRTSYSRPT (Q)[4]	QPFR/QPTTNSRP *PGM	*USE	*EXECUTE
	QAPMDMPT *FILE		*EXECUTE
	Job description	*USE	*EXECUTE
PRTTNSRPT (Q)[4]	QPFR/QPTTNSRP *PGM	*USE	*EXECUTE
	Trace file (QTRJOBT) library		*EXECUTE
	Job description	*USE	*EXECUTE
PRTTRCRPT (Q)[4]	QPFR/QPTTRCCP *PGM	*USE	*EXECUTE
RMVPEXDFN (Q)[5]			
RMVPEXFTR (Q)[5]			
STRBEST (Q)[4]	QPFR/QCYBMAIN *PGM	*USE	*EXECUTE
STRDBMON[3, 4]	Output file	*OBJOPR, *ADD	*EXECUTE
STRJOBTRC (Q)	QPFR/QPTTRCJ1 *PGM	*USE	*EXECUTE
STRPEX (Q)[5]			
STRPFRG (Q)[4]	QPFR/QPGSTART *PGM	*USE	*EXECUTE

Performance Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
STRPFRT (Q)[4]	QPFR/QMNMAIN0 *PGM	*USE	*EXECUTE
	QAPTAPGP *FILE in the functional areas library	*CHANGE	*EXECUTE
	CHGFCNARA command (Q)	*USE	*EXECUTE
	CPYFCNARA command (Q)	*USE	*EXECUTE
	CRTFCNARA command (Q)	*USE	*EXECUTE
	DLTFCNARA command (Q)	*USE	*EXECUTE
	QPFR/QPTAGRP *PGM	*USE	*EXECUTE
	QPFR/QPTAGRPD *PGM	*USE	*EXECUTE
	QPFR/QPTAGRPR *PGM	*USE	*EXECUTE
WRKFCNARA (Q)[4]	QPFR/QPTAGRPC *PGM	*USE	*EXECUTE
	Output file (QAITMON)	*CHANGE, *ALTER	*EXECUTE, *ADD
WRKPEXDFN (Q)[5]			
WRKPEXFTR (Q)[5]			
WRKSYSACT (Q)[3, 4]	QPFR/QITMONCP *PGM	*USE	*EXECUTE
These commands do not require any object authorities: • ENDDBMON[3] • ENDPFRTRC (Q) • STRPFRTRC (Q)			

[1] If the default library (QPEXDATA) is specified, authority to that library is not checked.

[2] Authority is needed to the library that contains the set of database files. Authority to the individual set of database files is not checked.

[3] To use this command, you must have *JOBCTL special authority.

[4] To use this command, you must have *SERVICE special authority.

[5] To use this command, you must have *SERVICE special authority or you must be authorized to the Service Trace function of Operating System/400 through iSeries Navigator's Application Administration support. The Change Function Usage Information (QSYCHFUI) API, with a function ID of QIBM_SERVICE_TRACE, can also be used to change the list of users that are allowed to perform trace operations..

Print Descriptor Group Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGPDGPRF	User profile	*OBJMGT	
CRTPDG	Print descriptor group		*READ, *ADD
DLTPDG	Print descriptor group	*OBJEXIST	*EXECUTE
DSPPDGPRF	User profile	*OBJMGT	
RTVPDGPRF	User profile	*READ	

Print Services Facility Configuration Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGPSFCFG [1, 2]			
CRTGPSFCFG [1, 2]			*READ, *ADD
DLTPSFCFG [1, 2]	PSF Configuration	*OBJEXIST	*EXECUTE
DSPPSFCFG [1]	PSF Configuration	*USE	*EXECUTE
WRKPSFCFG [1]	PSF Configuration	*READ	*EXECUTE

[1] The PSF/400 feature is required to use this command.

[2] *IOSYSCFG special authroity is required to use this command.

Problem Commands

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	For Library
ADDPRBACNE (Q)	Filter	*USE, *ADD	*EXECUTE
ADDPRBSLTE (Q)	Filter	*USE, *ADD	*EXECUTE
ANZPRB (Q)	SNDSRVRQS command	*USE	*EXECUTE
CHGPRB (Q)			*EXECUTE
CHGPRBACNE (Q)	Filter	*USE, *UPD	*EXECUTE
CHGPRBSLTE (Q)	Filter	*USE, *UPD	*EXECUTE
DLTPRB (Q) [3]	Command: DLTAPARDTA	*USE	*EXECUTE
DSPPRB	Output file	See General Rules on page 311	See General Rules on page 311
PTRINTDTA (Q)			
QRYPRBSTS (Q)			
VFYCMN (Q)	Line description [1]	*USE	*EXECUTE
	Controller description [1]	*USE	*EXECUTE
	Network ID [1]	*USE	*EXECUTE
VFYOPT (Q)	Device description	*USE	*EXECUTE
VFYTAP (Q)	Device description	*USE	*EXECUTE
VFYPRT (Q)	Device description	*USE	*EXECUTE
WRKPRB (Q) [2]	Line, controller, NWID (Netword ID), and device based on problem analysis action	*USE	*EXECUTE

[1] You need *USE authority to the communications object you are verifying.

[2] You must have *USE authority to the SNDSRVRQS command to be able to report a problem.

[3] You must have authority to DLTAPARDTA if you want the APAR data associated with the problem to be deleted also. See DLTAPARDTA in the Service Commands-Authorities Needed table to determine additional authorities that are needed.

Program Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
The object authorities required for the CRTxxxPGM commands are listed in the Languages table in "Language Commands" on page 376			
ADDBKP [1]	Breakpoint handling program	*USE	*EXECUTE
ADDPGM [1,2]	Program	*CHANGE	*EXECUTE
ADDTRC [1]	Trace handling program	*USE	*EXECUTE
CALL	Program	*OBJOPR, *EXECUTE	*EXECUTE
	Service program [4]	*EXECUTE	*EXECUTE
CHGDBG	Debug operation	*USE, *ADD, *DLT	*EXECUTE
CHGHLLPTR [1]			
CHGPGM	Program	*OBJMGT, *USE	*USE
	Program, if recreate option specified, optimization level changed, or performance data collection changed	*OBJMGT, *USE	*USE, *ADD, *DLT
	Program, if USRPRF or USEADPAUT parameter is being changed	Owner [7]	*USE, *ADD, *DLT
CHGPGMVAR [1]			
CHGPTR [1]			
CHGSRVPGM	Service program	*OBJMGT, *USE	*USE
	Service program, if recreate option specified, optimization level changed, or performance data collection changed	*OBJMGT, *USE	*USE, *ADD, *DLT
	Service program, if USRPRF or USEADPAUT parameter is being changed.	Owner [7], *USE, *OBJMGT	*USE, *ADD, *DLT
CLRTRCDTA [1]			
CRTPGM	Program, Replace(*NO)	See General Rules on page 311	*READ, *ADD
	Program, Replace(*YES)	See General Rules on page 311	*READ, *ADD
	Service program specified in the BNDSRVPGM parameter.	*USE	*EXECUTE
	Module	*USE	*EXECUTE
	Binding directory	*USE	*EXECUTE
CRTSRVPGM	Service program, Replace(*NO)	See General Rules on page 311	*READ, *ADD
	Service program, Replace(*YES)	See General Rules on page 311	*READ, *ADD
	Module	*USE	*EXECUTE
	Service program specified in BNDSRVPGM parameter	*USE	*EXECUTE
	Export source file	*OBJOPR *READ	*EXECUTE
	Binding directory	*USE	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
CVTCLSRC	From-file	*USE	*EXECUTE
	To-file	*OBJOPR, *OBJMGT, *USE, *ADD, *DLT	*READ, *ADD
DLTDFUPGM	Program	*OBJEXIST	*EXECUTE
	Display file	*OBJEXIST	*EXECUTE
DLTPGM	Program	*OBJEXIST	*EXECUTE
DLTSRVPGM	Service program	*OBJEXIST	*EXECUTE
DMPCLPGM	CL Program	*USE	None [3]
DSPBKP [1]			
DSPDBG [1]			
DSPDBGWCH			
DSPMODSRC[2, 4]	Source file	*USE	*USE
	Any include files	*USE	*USE
	Program	*CHANGE	*EXECUTE
DSPPGM	Program	*READ	*EXECUTE
	Program, if DETAIL(*MODULE) specified	*USE	*EXECUTE
DSPPGMREF	Program	*OBJOPR	*EXECUTE
	Output file	See General Rules on page 311	See General Rules on page 311
DSPPGMVAR [1]			
DSPSRVPGM	Service program	*READ	*EXECUTE
	Service program, if DETAIL(*MODULE) specified	*USE	*EXECUTE
DSPTRC [1]			
DSPTRCDTA [1]			
ENDCBLDBG (COBOL/400 licensed program or S/38 environment)	Program	*CHANGE	*EXECUTE
ENDDBG [1]	Source debug program	*USE	*USE
ENDRQS [1]			*EXECUTE
ENTCBLDBG (S/38 environment)	Program	*CHANGE	*EXECUTE
EXTPGMINF	Source file and database files	*OBJOPR	*EXECUTE
	Program information		*READ, *ADD
PRTCMDUSG	Program	*USE	*EXECUTE
RMVBKP [1]			
RMVPGM [1]			
RMVTRC [1]			
RSMBKP [1]			

Program Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
RTVCLSRC	Program	*OBJMGT, *USE	*EXECUTE
	Database source file	*OBJOPR, *OBJMGT, *ADD, *DLT	*EXECUTE
SETATNPGM	Attention-key-handling program	*OBJOPR or one or more data authorities	*EXECUTE
SETPGMINF	Database files	*OBJOPR	*EXECUTE
	Source file	*USE	*EXECUTE
	Root program	*CHANGE	*READ, *ADD
	Sub-program	*USE	*EXECUTE
STRCBLDBG	Program	*CHANGE	*EXECUTE
STRDBG	Program [2]	*CHANGE	*EXECUTE
	Source file [4]	*USE	*EXECUTE
	Any include files [4]	*USE	*EXECUTE
	Source debug program	*USE	*EXECUTE
	Unmonitored message program	*USE	*EXECUTE
TFRCTL [4]	Program	*USE or a data authority other than *EXECUTE	*EXECUTE
	Some language functions when using high-level languages	*READ	*EXECUTE
UPDPGM	Program	*OBJMGT, *OBJEXIST, *USE	*USE, *ADD
	Service program specified in the BNDSRVPGM parameter.	*USE	*EXECUTE
	Module	*USE	*EXECUTE
	Binding directory	*USE	*EXECUTE
UPDSRVPGM	Service Program	*OBJMGT, *OBJEXIST, *USE	*USE, *ADD
	Service program specified in BNDSRVPGM parameter	*USE	*EXECUTE
	Module	*USE	*EXECUTE
	Binding directory	*USE	*EXECUTE
	Export source file	*OBJOPR *READ	*EXECUTE
WRKPGM [6]	Program	Any authority	*USE
WRKSRVPGM [6]	Service program	Any authority	*USE

Command	Referenced Object	Authority Needed	
		For Object	For Library
[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 DB2 Universal Database for iSeries topic in the Information Center contains more information about security requirements for SQL statements. See "Prerequisite and related information" on page xvi for details.			
[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.			

Query Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
ANZQRY	Query definition	*USE	*EXECUTE
CHGQRYA[4]			
CRTQMFORM	Query management form: REPLACE(*NO)		*READ, *ADD, *EXECUTE
	Query management form: REPLACE(*YES)	*ALL	*READ, *ADD, *EXECUTE
	Source file	*USE	*EXECUTE
CRTQMQRY	Query management query: REPLACE(*NO)		*READ, *ADD, *EXECUTE
	Query management query: REPLACE(*YES)	*ALL	*READ, *ADD, *EXECUTE
	Source file	*USE	*EXECUTE
	OVRDBF command	*USE	*EXECUTE
DLTQMFORM	Query management form	OBJEXIST	*EXECUTE
DLTQMQRY	Query management query	*OBJEXIST	*EXECUTE
DLTQRY	Query definition	*OBJEXIST	*EXECUTE
RTVQMFORM	Query manager form	*OBJEXIST	*EXECUTE
	Target source file	*ALL	*READ, *ADD, *EXECUTE
	ADDPFM, CHGPFM, CLRPFM, CPYSRCF, CRTPRTF, CRTSRCPF, DLTF, DLTOVR, OVRDBF, RMVM commands	*USE	*EXECUTE
RTVQMQRY	Query manager query	*USE	*EXECUTE
	Target source file	*ALL	*READ, *ADD
	ADDPFM, CHGPFM, CLRPFM, CPYSRCF, CRTPRTF, CRTSRCPF, DLTF, DLTOVR, OVRDBF, RMVM commands	*USE	*EXECUTE

Query Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
RUNQRY	Query definition	*USE	*USE
	Input files	*USE	*EXECUTE
	Output files	See General Rules on page 311	See General Rules on page 311
STRQMQRY [1]	Query management query	*USE	*EXECUTE
	Query management form, if specified	*USE	*EXECUTE
	Query definition, if specified	*USE	*EXECUTE
	Output file	See General Rules on page 311	See General Rules on page 311
	ADDPFM, CHGOBJD, CHGPFM, CLRPFM, CPYSRCF, CRTPRTF, CRTSRCPF, DLTF, DLTOVR, GRTOBJAUT OVRDBF, OVRPRTF RMVM commands (if OUTPUT(*OUTFILE) is specified)	*USE	*EXECUTE
STRQMPRC [1]	Source file containing query manager procedure	*USE	*EXECUTE
	Source file containing command source file, if specified	*USE	*EXECUTE
	OVRPRTF command, if statements result in printed report or query object.	*USE	*EXECUTE
STRQRY			*EXECUTE
WRKQMFORM [3]	Query management form	Any authority	*USE
WRKQMQRY [3]	Query management query	Any authority	*USE
WRKQRY [3]			

[1] To run STRQM, you must have the authority required by the statements in the query. For example, to insert a row in a table requires *OBJOPR, *ADD, and *EXECUTE 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.

[4] To use individual command, you must have *JOBCTL special authority.

QSH Shell Interpreter Commands

These commands do not require any authorities to objects:
STRQSH [1] QSH[1]

[1] QSH is an alias for the STRQSH CL command.

Question and Answer Commands

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	For Library
ANSQST (Q)	Database file QAQAxxBQPY [1]	*READ	*READ
ASKQST	Database file QAQAxxBBPY [1] or QAQAxxBQPY [1]	*READ	*READ
CHGQSTDB (Q)	Database file QAQAxxBQPY [1]	*READ	*READ
CRTQSTDB [2] (Q)	Database files		*READ, *ADD, *EXECUTE
CRTQSTLOD (Q)	Database file QAQAxxBQPY [1]	*READ	*READ
DLTQST (Q)	Database file QAQAxxBQPY [1]	*READ	*READ
DLTQSTDB (Q)	Database file QAQAxxBQPY [1]	*READ	*READ
EDTQST (Q)	Database file QAQAxxBQPY [1]	*READ	*READ
LODQSTDB [2] (Q)	Database file QAQAxxBQPY [1,3]	*READ	*READ, *ADD, *EXECUTE
STRQST [4]	Database file QAQAxxBBPY [1] or QAQAxxBQPY [1]	*READ	*READ
WRKQST	Database file QAQAxxBBPY [1] QAQAxxBQPY [1]	*READ	*USE
WRKCNTINF			*EXECUTE

[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 Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
STRDBRDR	Message queue	*OBJOPR, *ADD	*EXECUTE
	Database file	*OBJOPR, *USE	*EXECUTE
	Job queue	*READ	*EXECUTE
STRDKTRDR	Message queue	*OBJOPR, *ADD	*EXECUTE
	Job queue	*READ	*EXECUTE
	Device description	*OBJOPR, *READ	*EXECUTE
These commands to not require any authority to objects:			
ENDRDR [1]	HLDRDR [1]	RLSRDR [1]	

[1] You must be the user who started the reader, or you must have all object (*ALLOBJ) or job control (*JOBCTL) special authority.

Registration Facility Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
ADDEXITPGM (Q)	Exit program	*USE	*EXECUTE
RMVEXITPGM (Q)	Exit program	*USE	*EXECUTE
WRKREGINF	Exit program	*USE	*EXECUTE

Relational Database Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
ADDRDBDIRE	Output file, if specified	*EXECUTE	*EXECUTE
CHGRDBDIRE	Output file, if specified	*EXECUTE	*EXECUTE
	Remote location devide description[7]	*CHANGE	
DSPRDBDIRE	Output file, if specified	See General Rules on page 311	See General Rules on page 311
These commands do not require any authority to objects:			
RMVRDBDIRE WRKRDBDIRE			
[1] Authority verified when the RDB directory entry is used.			

Resource Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
DSPHDWRSC			
DSPSFWRSC	Output file, if specified	See General Rules on page 311	See General Rules on page 311
EDTDEVRSC			
WRKHDWRSC [1]			
[1] If you use the option to create a configuration object, you must have authority to use the appropriate CRT command.			

RJE (Remote Job Entry) Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
ADDFCTE	Forms control table	*DELETE, *USE, *ADD	*READ, *EXECUTE
	Device file [1,2]	*USE	*READ, *EXECUTE
	Physical file [1,2] (RJE generates members)	*OBJMGT, *USE, *ADD	*READ, *EXECUTE, *ADD
	Physical file [1,2] (member specified)	*USE, *ADD	*READ, *EXECUTE
	Program [1,2]	*USE	*READ, *EXECUTE
	Message queue [1,2]	*USE, *ADD	*READ, *EXECUTE
	QUSER user profile	*USE	*READ, *EXECUTE
ADDRJECMNE	Session description	*USE, *ADD, *DLT	*READ, *EXECUTE
	BSC/CMN file [1,2]	*USE	*READ, *EXECUTE
	Device description [2]	*USE	*READ, *EXECUTE
	QUSER user profile	*USE	*READ, *EXECUTE
ADDRJERDRE	Session description	*READ, *ADD, *DLT	*READ, *EXECUTE
	Job queue [2]	*READ	*READ, *EXECUTE
	Message queue [2]	*READ, *ADD	*READ, *EXECUTE
ADDRJEWTRE	Session description	*READ, *ADD, *DLT	*READ, *EXECUTE
	Device file [1,2]	*USE	*READ, *EXECUTE
	Physical file [1,2] (RJE generates members)	*OBJMGT, *USE, *ADD	*READ, *EXECUTE, *ADD
	Physical file [1,2] (member specified)	*OBJOPR, *ADD	*READ, *EXECUTE
	Program [1,2]	*USE	*READ, *EXECUTE
	Message queue [1,2]	*USE, *ADD	*READ, *EXECUTE
	QUSER user profile	*USE	*READ, *EXECUTE
CHGFCT	Forms control table	*OBJOPR, *OBJMGT	*READ, *EXECUTE
CHGFCTE	Forms control table	*USE	*READ, *EXECUTE
	Device file [1,2]	*USE	*READ, *EXECUTE
	Physical file [1,2] (RJE generates members)	*OBJMGT, *USE, *ADD	*READ, *EXECUTE, *ADD
	Physical file [1,2] (member specified)	*USE, *ADD	*READ, *EXECUTE
	Program [1,2]	*USE	*READ, *EXECUTE
	Message queue [1,2]	*USE, *ADD	*READ, *EXECUTE
	QUSER user profile	*USE	*READ, *EXECUTE
CHGRJECMNE	Session description	*USE	*READ, *EXECUTE
	BSC/CMN file [1,2]	*USE	*READ, *EXECUTE
	Device description [2]	*USE	*READ, *EXECUTE
	QUSER user profile	*USE	*READ, *EXECUTE

RJE (Remote Job Entry) Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGRJERDRE	Session description	*USE, *ADD, *DLT	*READ, *EXECUTE
	Job queue [2]	*USE	*READ, *EXECUTE
	Message queue [2]	*USE, *ADD	*READ, *EXECUTE
CHGRJEWTRE	Session description	*USE	*READ, *EXECUTE
	Device File [1,2]	*USE	*READ, *EXECUTE
	Physical file [1,2] (RJE generates members)	*OBJMGT, *USE, *ADD	*READ, *EXECUTE, *ADD
	Physical file [1,2] (member specified)	*OBJOPR, *ADD	*READ, *EXECUTE
	Program [1,2]	*USE	*READ, *EXECUTE
	Message queue [1,2]	*USE, *ADD	*READ, *EXECUTE
	QUSER user profile	*USE	*READ, *EXECUTE
CHGSSND	Session description	*OBJMGT, *READ, *UPD, *OBJOPR	*EXECUTE, *READ
	Job queue [1,2]	*USE	*EXECUTE
	Message queue [1,2]	*USE, *ADD	*EXECUTE
	Forms control table [1,2]	*USE	*EXECUTE
	QUSER user profile	*USE	*EXECUTE
CNLRJERDR	Session description	*USE	*EXECUTE
	Message queue	*USE, *ADD	*EXECUTE
CNLRJEWTR	Session description	*USE	*EXECUTE
	Message queue	*USE, *ADD	*EXECUTE
CRTFCT	Forms control table		*READ, *ADD
CRTRJEBSCF	BSC file		*READ, *EXECUTE, *ADD
	Source physical file (DDS)	*READ	*EXECUTE
	Device description	*READ	*EXECUTE
CRTRJECFG	Session description		*READ, *ADD, *UPD, *OBJOPR
	Job queue		*READ, *ADD
	Job description		*READ, *OBJOPR, *ADD
	Subsystem description		*READ, *OBJOPR, *ADD
	Message queue		*READ, *ADD
	CMN file		*READ, *EXECUTE, *ADD
	BSC file		*READ, *EXECUTE, *ADD
	Printer file		*USE, *ADD

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTRJECFG (continued)	Physical file		*EXECUTE, *ADD
	User profile QUSER [3]	*USE	*EXECUTE
	Output queue	*READ	*EXECUTE
	Forms control table	*READ	*READ
	Device description		*EXECUTE
	Controller description		*EXECUTE
	Line description		*EXECUTE
CRTRJECMNF	Communication file		*READ, *EXECUTE, *ADD
	Source physical file (DDS)	*READ	*EXECUTE
	Device description	*READ	*EXECUTE
CRTSSND	Session description		*READ, *ADD, *UPD, *OBJOPR
	Job queue [1,2]	*USE	*EXECUTE
	Message queue [1,2]	*USE, *ADD	*EXECUTE
	Forms control table [1,2]	*USE	*EXECUTE
	QUSER user profile	*USE	*EXECUTE
CVTRJEDTA	Forms control table	*USE	*EXECUTE
	Input file	*USE, *UPD	*EXECUTE
	Output file (RJE generates member)	*OBJMGT, *USE, *ADD	*READ, *EXECUTE, *ADD
	Output file (member specified)	*USE, *ADD	*EXECUTE
DLTFCT	Forms control table	*OBJEXIST	*EXECUTE
DLTRJECFG	Session description	*OBJEXIST	*EXECUTE
	Job queue	*OBJEXIST	*EXECUTE
	BSC/CMN file	*OBJEXIST, *OBJOPR	*EXECUTE
	Physical file	*OBJEXIST, *OBJOPR	*EXECUTE
	Printer file	*OBJEXIST, OBJOPR	*EXECUTE
	Message queue	*OBJEXIST, *USE, *DLT	*EXECUTE
	Job description	*OBJEXIST	*EXECUTE
	Subsystem description	*OBJEXIST, *USE	*EXECUTE
	Device description [4]	*OBJEXIST	*EXECUTE
	Controller description [4]	*OBJEXIST	*EXECUTE
	Line description [4]	*OBJEXIST	*EXECUTE
DLTSSND	Session description	*OBJEXIST	*EXECUTE
DSPRJECFG	Session description	*READ	*EXECUTE
ENDRJESSN [5]	Session description	*USE	*EXECUTE
RMVFCTE	Forms control table	*OBJOPR, *READ, *ADD, *DLT	*EXECUTE

RJE (Remote Job Entry) Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
RMVRJECMNE	Session description	*OBJOPR, *READ, *ADD, *DLT	*EXECUTE
RMVRJERDRE	Session description	*OBJOPR, *READ, *ADD, *DLT	*EXECUTE
RMVRJEWTRE	Session description	*OBJOPR, *READ, *ADD, *DLT	*EXECUTE
SNDRJECMD	Session description	*USE	*EXECUTE
SBMRJEJOB	Session description	*USE	*EXECUTE
	Input file [6]	*USE	*EXECUTE
	Message queue	*USE, *ADD	*EXECUTE
	Job-related objects [7]		
SNDRJECMD	Session description	*USE	*EXECUTE
STRRJECSL	Session description	*USE	*EXECUTE
	Message queue	*USE	*EXECUTE
STRRJERDR	Session description	*USE	*USE
STRRJESSN [5]	Session description	*USE	*USE, *ADD
	Program	*USE	*EXECUTE
	User profile QUSER	*USE	*EXECUTE
	Job-related objects [7]		*EXECUTE
STRRJEWTR	Session description	*USE	*USE
	Program [1]	*USE	*READ, *EXECUTE
	Device file [1]	*USE, *ADD	*READ, *EXECUTE
	Physical file [1] (RJE generates members)	*OBJMGT, *USE, *ADD	*OBJOPR, *ADD
	Physical file [1] (member specified)	*READ, *ADD	*READ, *EXECUTE
	Message queue [1]	*USE, *ADD	*READ, *EXECUTE
	QUSER user profile	*USE	*READ, *EXECUTE
WRKFCT [8]	Forms control table	*USE	*EXECUTE
WRKRJESSN [8]	Session description	*USE	*EXECUTE
WRKSSND [8]	Session description	*CHANGE	*EXECUTE

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 370.

8 To use an individual operation, you must have the authority required by the operation.

Security Attributes Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGSECA [1]			
CHGSECAUD [2,3]			
CFGSYSSEC [1,2,3]			
DSPSECA			
DSPSECAUD [2,3]			
PRTSYSSECA [2]			

[1] You must have *SECADM special authority to use this command.

[2] You must have *ALLOBJ special authority to use this command.

[3] You must have *AUDIT special authority to use this command.

Server Authentication Entry Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
ADDSVRAUTE[1]			
CHGSVRAUTE[1]			
DSPSVRAUTE	User profile	*READ	*EXECUTE
RMVSVRAUTE[1]			

[1] If the user profile for this operation is not *CURRENT or the current user for the job, you must have *SECADM special authority and *OBJMGT and *USE authority to the profile.

Service Commands

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	For Library
APYPTF (Q)	Product library	*OBJMGT	
CHGSRVA [3] (Q)			
CHKCMNTRC [3] (Q)			*EXECUTE
CHKPRDOPT (Q)	All objects in product option [4]		

Service Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CPYPTF[2](Q)	From file	*USE	*EXECUTE
	To-file[8]	Same requirements as the SAVOBJ command	Same requirements as the SAVOBJ command
	Device description	*USE	*EXECUTE
	Licensed program		*USE
	Commands: CHKTAP, CPYFRMTAP, CPYTOTAP, CRTLIB, CRTSAVF, CRTTAPF, and OVRTAPF	*USE	*EXECUTE
	QSRV library	*USE	*EXECUTE
CPYPTFGRP[2] (Q)	Device description	*USE	*EXECUTE
	To-file	*Same requirements as the SAVOBJ command	*Same requirements as the SAVOBJ command
	From-file	*USE	*EXECUTE
	Commands: CHKTAP, CRTLIB, CRTSAVF	*USE	*EXECUTE
DLTAPARDTA (Q)			
DLTCMNTRC [3] (Q)	NWID (network ID) or line description	*USE	*EXECUTE
DLTPTF (Q)	Cover letter file [4]		*EXECUTE
	PTF save file [4]		*EXECUTE
DLTTRC (Q)	RMVM command	*USE	
	QSYS Library	*EXECUTE	
	Database Files	*OBJEXIST, *OBJOPR	
DMPJOB (Q)			*EXECUTE
DMPJOBINT (Q)			
DSPPTF (Q)	Output file	See General Rules on page 311	See General Rules on page 311
DSPSRVA (Q)			
DSPSRVSTS (Q)			
ENDCMNTRC [3] (Q)	NWID or line description	*USE	*EXECUTE
ENDCPYSCN (Q)	Device description	*USE	*EXECUTE
ENDSRVJOB (Q)			
ENDTRC (Q)	QSYS Library	*ADD, *EXECUTE	
	Database files	*OBJOPR, *OBJMGMT, *ADD, *DLT	
	Commands: PTRTRC, DLTTRC	*USE	
INSPTF[9] (Q)			
LODPTF (Q)	Device Description	*USE	*EXECUTE
LODRUN [2]	RSTOBJ command	*USE	*EXECUTE
PRTCMNTRC [3] (Q)	NWID (network ID) or line description	*USE	*EXECUTE
	Output file	See General Rules on page 311	See General Rules on page 311

Command	Referenced Object	Authority Needed	
		For Object	For Library
PRTERRLOG (Q)	Output file	See General Rules on page 311	See General Rules on page 311
PRTINTDTA (Q)			
PRTTRC (Q)	QSYS Library	*EXECUTE	
	Database Files	*USE	
	DLTTRC command	*USE	
RMVPTF (Q)	Product library	*OBJMGT	
RUNLPDA (Q)	Line description	*READ	*EXECUTE
SAVAPARDTA [6] (Q)	Commands: CRTDUPOBJ, CRTLIB, CRTOUTQ, CRTSAVF, DLTF, DMPOBJ, DMPSYSOBJ, DSPCTLD, DSPDEVD, DSPHDWRSC, DSPJOB, DSPLIND, DSPLOG, DSPNWID, DSPPTF, DSPSFWRSC, OVRPRTF, PRTERRLOG, PRTINTDTA, SAV, SAVDLO, SAVLIB, SAVOJB, WRKACTJOB, and WRKSYSVAL	*USE	*EXECUTE
	Existing problem [7]	*CHANGE	*EXECUTE
SNDPTFORD [10] (Q)			
SNDSRVRQS (Q)			
STRCMNTRC [3] (Q)	NWID (network ID) or line description	*USE	*EXECUTE
STRCPYSCN	Job queue	*USE	*EXECUTE
	Device description	*USE	*EXECUTE
	Output file, if specified	See General Rules on page 311	See General Rules on page 311
STRSRVJOB (Q)	User profile of job	*USE	*EXECUTE
STRSST [3] (Q)			
STRTRC (Q)		*READ, *WRITE	
TRCCNN[11]			
TRCCPIC (Q)			
TRCICF (Q)			
TRCINT[11] (Q)			
TRCJOB (Q)	Output file, if specified	See General Rules on page 311	See General Rules on page 311
	Exit program, if specified	*USE	*EXECUTE
VFYCMN (Q)	Line description [5]	*USE	*EXECUTE
	Controller description [5]	*USE	*EXECUTE
	Network ID [5]	*USE	*EXECUTE
VFYLNKLPDA (Q)	Line description	*READ	*EXECUTE
VFYPRT (Q)	Device description	*USE	*EXECUTE
VFYOPT (Q)	Device description	*USE	*EXECUTE
VFYTAP (Q)	Device description	*USE	*EXECUTE
WRKCNTINF (Q)			
WRKFSTAF (Q)	QUSRSYS/QPVINDEX *USRIDX	*CHANGE	*USE

Service Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
WRKFSTPCT (Q)	QUSRSYS/QPVPCTABLE *USRIDX	*CHANGE	*USE
WRKPRB [1, 10] (Q)	Line, controller, NWID (Network ID), and device based on problem analysis action	*USE, *ADD	*EXECUTE
WRKPTFGRP (Q)			
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 that 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.

9 This command has the same authorities and restrictions as the APYPTF command and the LODPTF command.

10 To access options 1 and 3 on the ″Select Reporting Option″ display, you must have *USE authority to the SNDSRVRQS command.

11 To use this command, you must have *SERVICE special authority, or be authorized to the Service Trace function of OS/400 through Operations Navigator's Application Administration support. The Change Function Usage Information (QSYCHFUI) API, with a function ID of QIBM_SERVICE_TRACE, can also be used to change the list of users that are allowed to perform trace operations.

11 To use this command, you must have *SERVICE special authority, or be authorized to the Service Trace function of OS/400 through Operations Navigator's Application Administration support. The Change Function Usage Information (QSYCHFUI) API, with a function ID of QIBM_SERVICE_TRACE, can also be used to change the list of users that are allowed to perform trace operations.

Spelling Aid Dictionary Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTSPADCT	Spelling aid dictionary	*OBJEXIST	*EXECUTE
	Dictionary - REPLACE(*NO)		*READ, *ADD
	Dictionary - REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
DLTSPADCT	Spelling aid dictionary	*OBJEXIST	*EXECUTE
WRKSPADCT [1]	Spelling aid dictionary	Any authority	*USE

1 To use an individual operation, you must have the authority required by the operation .

Sphere of Control Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
ADDSOCE	Sphere of control [1]	*USE, *ADD	*EXECUTE
DSPSOCSTS			
RMVSOCE	Sphere of control [1]	*USE, *DLT	*EXECUTE
WRKSOC	Sphere of control [1]	*USE	*EXECUTE

1 The sphere of control is physical file QUSRSYS/QAALSOC.

Spooled File Commands

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		Output Queue Parameters			Special Authority
		For Object	For Library	DSPDTA	AUTCHK	OPRCTL	
CHGSPLFA [1,2]	Output queue [3]	*READ, *DLT, *ADD	*EXECUTE		*DTAAUT		
		Owner [4]	*EXECUTE		*OWNER		
						*YES	*JOBCTL
CHGSPLFA [1], if moving spooled file	Original output queue [3]	*READ, *ADD, *DLT	*EXECUTE		*DTAAUT		
		Owner [4]	*EXECUTE		*OWNER		
				*YES or *NO		*YES	*JOBCTL
	Spooled file	Owner		*OWNER			
	Target output queue[7]	*READ	*EXECUTE			*YES	*JOBCTL
	Target device	*USE					
CPYSPLF [1]	Database file	See General Rules on page 311	See General Rules on page 311				
	Spooled file	Owner		*OWNER			
	Output queue [3]	*READ	*EXECUTE	*YES			
		*READ, *ADD, *DLT	*EXECUTE	*NO	*DTAAUT		
		Owner [4]	*EXECUTE	*NO	*OWNER		
			*EXECUTE	*YES or *NO		*YES	*JOBCTL
DLTSPLF [1]	Output queue [3]	*READ, *ADD, *DLT	*EXECUTE		*DTAAUT		
		Owner [4]	*EXECUTE		*OWNER		
						*YES	*JOBCTL

Spooled File Commands

Command	Referenced Object	Authority Needed		Output Queue Parameters			Special Authority
		For Object	For Library	DSPDTA	AUTCHK	OPRCTL	
DSPSPLF [1]	Output queue [3]	*READ	*EXECUTE	*YES			
		*READ, *ADD, *DLT	*EXECUTE	*NO	*DTAAUT		
		Owner [4]	*EXECUTE	*NO	*OWNER		
				*YES or *NO		*YES	*JOBCTL
	Spooled file	Owner		*OWNER			
HLDSPLF [1]	Output queue [3]	*READ, *ADD, *DLT	*EXECUTE		*DTAAUT		
		Owner [4]	*EXECUTE		*OWNER		
						*YES	*JOBCTL
RCLSPLSTG (Q)							
RLSSPLF [1, 8]	Output queue [3]	*READ, *ADD, *DLT	*EXECUTE		*DTAAUT		
		Owner [4]	*EXECUTE		*OWNER		
						*YES	*JOBCTL
SNDNETSPLF [1,5]	Output queue [3]	*READ	*EXECUTE	*YES			
		*READ, *ADD, *DLT	*EXECUTE	*NO	*DTAAUT		
		Owner [4]	*EXECUTE	*NO	*OWNER		
			*EXECUTE	*YES or *NO		*YES	*JOBCTL
	Spooled file	Owner		*OWNER			
WRKSPLF							

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.

6 If you have job control (*JOBCTL) special authority and the output queue is set to OPRCTL(*YES), you do not need *EXECUTE authority to the library of the output queue.

7 If you have *SPLCTL special authority, you must have *EXECUTE authority to the target output queue library.

8 When the spooled file has been held with HLDJOB SPLFILE(*YES) and the spooled file was also decoupled from the job, the user will need to have *USE authority to the RLSJOB command and either have *JOBCTL special authority or be the owner of the spooled file.

Subsystem Description Commands

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	For Library
ADDAJE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
	Job description	*OBJOPR, *READ	*EXECUTE
ADDCMNE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
	Job description	*OBJOPR, *READ	*EXECUTE
	User profile	*OBJOPR, *READ	
ADDJOBQE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
ADDPJE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
	User profile for the program start request to specify *PGMSTRRQS	*OBJOPR, *READ	*EXECUTE
	User profile	*OBJOPR, *READ	
	Job description	*OBJOPR, *READ	*EXECUTE
ADDRTGE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
ADDWSE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
	Job description	*OBJOPR, *READ	*EXECUTE
CHGAJE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
	Job description	*OBJOPR, *READ	*EXECUTE
CHGCMNE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
	Job description	*OBJOPR, *READ	*EXECUTE
	User profile	*OBJOPR, *READ	
CHGJOBQE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
CHGPJE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
	User profile for the program start request to specify *PGMSTRRQS	*OBJOPR, *READ	*EXECUTE
	User profile	*OBJOPR, *READ	
	Job description	*OBJOPR, *READ	*EXECUTE
CHGRTGE	Subsystem description	*OJBOPR, *OBJMGT, *READ	*EXECUTE
CHGSBSD [5]	Subsystem description	*OJBOPR, *OBJMGT, *READ	*EXECUTE
	Sign-on display file[4]	*YES	*EXECUTE
CHGWSE	Subsystem description	*OJBOPR, *OBJMGT, *READ	*EXECUTE
	Job description	*OBJOPR, *READ	*EXECUTE

Subsystem Description Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTSBSD [5] (Q)	Subsystem description		*READ, *ADD
	Sign-on display file [4]	*USE	*EXECUTE
DLTSBSD	Subsystem description	*OBJEXIST, *USE	*EXECUTE
DSPSBSD	Subsystem description	*OBJOPR, *READ	*EXECUTE
ENDSBS [1]			
PRTSBSDAUT [6]			
RMVAJE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
RMVCMNE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
RMVJOBQE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
RMVPJE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
RMVRTGE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
RMVWSE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
STRSBS [1]	Subsystem description	*USE	*READ, *EXECUTE
WRKSBS [2, 3]	Subsystem description	Any authority	*USE
WRKSBSD [3]	Subsystem description	Any authority	*USE

1 You must have job control (*JOBCTL) special authority to use this command.

2 Requires some authority (anything but *EXCLUDE)

3 To use an individual operation, you must have the authority required by the operation.

4 The authority is needed to complete format checks of the display file. This helps predict that the display will work correctly when the subsystem is started. When you are not authorized to the display file or its library, those format checks will not be performed.

5 You must have *SECADMIN or *ALLBOJ special authority to specify a specific library for the subsystem library.

6 You must have *ALLBOJ special authority to use this command.

System Commands

These commands do not require any object authorities:

CHGSHRPOOL	RCLRSC	SIGNOFF	WRKSYSSTS
DSPSYSSTS	RETURN	WRKSHRPOOL	
ENDSYS[1]	RTVGRPA		
PWRDWNSYS[1]			
RCLACTGRP[1]			

1 You must have job control (*JOBCTL) special authority to use this command.

System Reply List Commands

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.

These commands do not require object authorities:			
ADDRPYLE (Q)	CHGRPYLE (Q)	RMVRPYLE (Q)	WRKRPYLE

System Value Commands

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.

These commands to not require any authority to objects:			
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 Commands

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.

		Authority Needed	
Command	**Referenced Object**	**For Object**	**For Library**
CHGS36	S/36 configuration object QS36ENV	*UPD	*EXECUTE
CHGS36A	S/36 configuration object QS36ENV	*UPD	*EXECUTE
CHGS36PGMA	Program	*OBJMGT, *USE	*EXECUTE
CHGS36PRCA	File QS36PRC	*OBJMGT, *USE	*EXECUTE
CHGS36SRCA	Source	*OBJMGT, *USE	*EXECUTE
CRTMSGFMNU	Menu: REPLACE(*NO)		*READ, *ADD
	Menu: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD
	Display file if it exists	*ALL	*EXECUTE
	Message file	*USE	*CHANGE
	Source file QS36SRC	*ALL	*EXECUTE
CRTS36DSPF	Display file: REPLACE(*NO)		*READ, *ADD
	Display file: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD, *CHANGE
	To-file source file when TOMBR is not *NONE	*ALL	*CHANGE
	Source file QS36SRC	*USE	*EXECUTE
	Create Display File (CRTDSPF) command	*OBJOPR	*EXECUTE

System/36 Environment Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTS36MNU	Menu: REPLACE(*NO)		*READ, *ADD, *CHANGE
	Menu: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD, *CHANGE
	To-file source file when TOMBR is not *NONE	*ALL	*CHANGE
	Source file QS36SRC	*USE	*EXECUTE
	Display file when REPLACE(*YES) is specified	*ALL	*EXECUTE
	Message files named in source	*ALL	*EXECUTE
	Display file		*CHANGE
	CRTMSGF command	*OBJOPR, *OBJEXIST	*EXECUTE
	ADDMSGD command	*OBJOPR	*EXECUTE
	CRTDSPF command	*OBJOPR	*EXECUTE
CRTS36MSGF	Message file: REPLACE(*NO)		*READ, *ADD, *CHANGE
	Message file: REPLACE(*YES)	See General Rules on page 311	*READ, *ADD, *CHANGE
	To-file source file when TOMBR is not *NONE	*ALL	*CHANGE
	Source file QS36SRC	*USE	*EXECUTE
	Display file when REPLACE(*YES) is specified	*ALL	*EXECUTE
	Message file named in source	*ALL	*EXECUTE
	Message file named in source when OPTION is *ADD or *CHANGE	*CHANGE	*EXECUTE
	Message files named in source when OPTION(*CREATE) is specified	*ALL	*EXECUTE
	CRTMSGF command	*OBJOPR, *OBJEXIST	*EXECUTE
	ADDMSGD command	*OBJOPR	*EXECUTE
	CHGMSGD command when OPTION(*CHANGE) is specified	*OBJOPR	*EXECUTE
DSPS36	S/36 configuration object QS36ENV	*READ	*EXECUTE
EDTS36PGMA	Program, to modify attributes	*OBJMGT, *USE	*EXECUTE
	Program, to view attributes	*USE	*EXECUTE
EDTS36PRCA	File QS36PRC, to modify attributes	*OBJMGT, *USE	*EXECUTE
	File QS36PRC, to view attributes	*USE	*EXECUTE
EDTS36SRCA	Source file QS36SRC, to modify attributes	*OBJMGT, *USE	*EXECUTE
	Source file QS36SRC, to view attributes	*USE	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
RSTS36F (Q)	From-file	*USE	*EXECUTE
	To-file	*ALL	See General Rules on page 311
	Based-on physical file, if file being restored is a logical (alternative) file	*CHANGE	*EXECUTE
	Device file or device description	*USE	*EXECUTE
RSTS36FLR [1,2,3] (Q)	S/36 folder	*USE	*EXECUTE
	To-folder	*CHANGE	*EXECUTE
	Device file or device description	*USE	*EXECUTE
RSTS36LIBM (Q)	From-file	*USE	*EXECUTE
	To-file	*ALL	See General Rules on page 311
	Device file or device description	*USE	*EXECUTE
RTVS36A	S/36 configuration object QS36ENV	*UPD	*EXECUTE
SAVS36F	From-file	*USE	*EXECUTE
	To-file, when it is a physical file	*ALL	See General Rules on page 311
	Device file or device description	*USE	*EXECUTE
SAVS36LIBM	From-file	*USE	*EXECUTE
	To-file, when it is a physical file	*ALL	See General Rules on page 311
	Device file or device description	*USE	*EXECUTE
WRKS36	S/36 configuration object QS36ENV	*READ	*EXECUTE
WRKS36PGMA	Program, to modify attributes	*OBJMGT, *USE	*EXECUTE
	Program, to view attributes	*USE	*EXECUTE
WRKS36PRCA	File QS36PRC, to modify attributes	*OBJMGT, *USE	*EXECUTE
	File QS36PRC, to view attributes	*USE	*EXECUTE
WRKS36SRCA	Source file QS36SRC, to modify attributes	*OBJMGT, *USE	*EXECUTE
	Source file QS36SRC, to view attributes	*USE	*EXECUTE

[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, 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.

Table Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTTBL	Table		*READ, *ADD, *EXECUTE
	Source file	*USE	*EXECUTE
DLTTBL	Table	*OBJEXIST	*EXECUTE

Table Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
WRKTBL [1]	Table	Any authority	*USE

[1] To use an individual operation, you must have the authority required by the operation .

TCP/IP Commands

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	For Library
CVTTCPCL (Q)	File objects	*USE	*EXECUTE
ENDTCP (Q)	Line description[4]	*USE	*EXECUTE
	Controller description[4]	*USE	*EXECUTE
	Device description[4]	*USE	*EXECUTE
	File Objects	*USE	*EXECUTE
ENDTCPIFC (Q)	File objects	*USE	*EXECUTE
	Line description[4]	*USE	*EXECUTE
	Controller description[4]	*USE	*EXECUTE
	Device description[4]	*USE	*EXECUTE
ENDTCPPTP	Line description[4]	*USE	*EXECUTE
	Controller description[4]	*USE	*EXECUTE
	Device description[4]	*USE	*EXECUTE
	File Objects	*USE	*EXECUTE
ENDTCPSRV (Q)	File objects	*USE	*EXECUTE
FTP	File objects	*USE	*EXECUTE
	Table objects	*USE	*EXECUTE
LPR [2]	Workstation customizing object	*USE	*EXECUTE
SETVTTBL	Table objects	*USE	*EXECUTE
SNDTCPSPLF [2]	Workstation customizing object	*USE	*EXECUTE
STRTCP (Q)	File objects	*USE	*EXECUTE
	Line description[4]	*USE	*EXECUTE
	Controller description[4]	*USE	*EXECUTE
	Device description[4]	*USE	*EXECUTE
STRTCPFTP	Table objects	*USE	*EXECUTE
	File objects	*USE	*EXECUTE
STRTCPIFC (Q)	File objects	*USE	*EXECUTE
	Line description[4]	*USE	*EXECUTE
	Controller description[4]	*USE	*EXECUTE
	Device description[4]	*USE	*EXECUTE

Command	Referenced Object	Authority Needed	
		For Object	For Library
STRTCPPTP	Line description[4]	*USE	*EXECUTE
	Controller description[4]	*USE	*EXECUTE
	Device description[4]	*USE	*EXECUTE
	File Objects	*USE	*EXECUTE
STRTCPSVR (Q)	Table objects	*USE	*EXECUTE
	File objects	*USE	*EXECUTE
STRTCPTELN	Table objects	*USE	*EXECUTE
	File objects	*USE	*EXECUTE
	Virtual workstation device[5]	*USE	*EXECUTE
TELNET	Table objects	*USE	*EXECUTE
	File objects	*USE	*EXECUTE
	Virtual workstation device[5]	*USE	*EXECUTE
These commands do not require any object authorities:			
ADDCOMSNMP [1]	CFGTCPSMTP	CHGVTMAP	RMVTCPRSI [1]
ADDNETTBLE [1]	CFGTCPSNMP	DSPVTMAP	RMVTCPRTE [1]
ADDPCLTBLE [1]	CFGTCPTELN	ENDTCPCNN	RNMTCPHTE [1]
ADDSRVTBLE [1]	CHGCOMSNMP [1]	MGRTCPHT [1]	SETVTMAP
ADDTCPHTE [1]	CHGFTPA [1]	NETSTAT	VFYTCPCNN
ADDTCPIFC [1]	CHGLPDA [1]	PING	WRKNAMSMTP [3]
ADDTCPPORT [1]	CHGSMTPA [1]	RMVCOMSNMP [1]	WRKNETTBLE[1]
ADDTCPRSI [1]	CHGSNMPA [1]	RMVNETTBLE [1]	WRKPCLTBLE[1]
ADDTCPRTE [1]	CHGTCPA [1]	RMVPCLTBLE [1]	WRKSRVTBLE[1]
CFGTCP	CHGTCPHTE[1]	RMVSRVTBLE [1]	WRKTCPSTS
CFGTCPAPP	CHGTCPIFC[1]	RMVTCPHTE[1]	
CFGTCPFTP [1]	CHGTCPRTE [1]	RMVTCPIFC[1]	
CFGTCPLPD [1]	CHGTELNA [1]	RMVTCPPORT [1]	

[1] You must have *IOSYSCFG special authority to use this command.

[2] The SNDTCPSPLF command and the LPR command use the same combinations of referenced object authorities as the SNDNETSPLF command. See page 428.

[3] You must have *SECADM special authority to change the system alias table or another user profile's alias table.

[4] If you have *JOBCTL special authority, you do not need the specified authority to the object.

[5] If you have *JOBCTL special authority, you do not need the specified authority to the object on the remote system.

Upgrade Order Information Data Commands

These commands 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	For Library
WRKORDINF	QGPL/QMAHFILE file	*CHANGE, *OBJALTER	*EXECUTE

User Index, User Queue, User Space Commands

Table 140.

Command	Referenced Object	Authority Needed	
		For Object	For Library
DLTUSRIDX	User index	*OBJEXIST	*EXECUTE
DLTUSRQ	User queue	*OBJEXIST	*EXECUTE
DLTUSRSPC	User space	*OBJEXIST	*EXECUTE

User Profile Commands

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	For Library
ANZDFTPWD [3, 14, 15](Q)			
ANZPRFACT [3, 14, 15](Q)			
CHGACTPRFL [14](Q)			
CHGACTSCDE [3, 14, 15](Q)			
CHGDSTPWD [1]			
CHGEXPSCDE [3, 14, 15](Q)			
CHGPRF	User profile	*OBJMGT, *USE	
	Initial program [2]	*USE	*EXECUTE
	Initial menu [2]	*USE	*EXECUTE
	Job description [2]	*USE	*EXECUTE
	Message queue [2]	*USE	*EXECUTE
	Output queue [2]	*USE	*EXECUTE
	Attention-key- handling program [2]	*USE	*EXECUTE
	Current library [2]	*USE	*EXECUTE
CHGPWD			
CHGUSRAUD [11](Q)			

Command	Referenced Object	Authority Needed	
		For Object	For Library
CHGUSRPRF [3]	User profile	*OBJMGT, *USE	*EXECUTE
	Initial program [2]	*USE	*EXECUTE
	Initial menu [2]	*USE	*EXECUTE
	Job description [2]	*USE	*EXECUTE
	Message queue [2]	*USE	*EXECUTE
	Output queue [2]	*USE	*EXECUTE
	Attention-key-handling program [2]	*USE	*EXECUTE
	Current library [2]	*USE	*EXECUTE
	Group profile (GRPPRF or SUPGRPPRF)[2,4]	*OBJMGT, *OBJOPR, *READ, *ADD, *UPD, *DLT	*EXECUTE
CHGUSRPRTI	User profile	*CHANGE	
CHKPWD			
CRTUSRPRF [3, 12, 17]	Initial program	*USE	*EXECUTE
	Initial menu	*USE	*EXECUTE
	Job description	*USE	*EXECUTE
	Message queue	*USE	*EXECUTE
	Output queue	*USE	*EXECUTE
	Attention-key- handling program	*USE	*EXECUTE
	Current library	*USE	*EXECUTE
	Group profile (GRPPRF or SUPGRPPRF)[4]	*OBJMGT, *OBJOPR, *READ, *ADD, *UPD, *DLT	*EXECUTE
DLTUSRPRF [3,9]	User profile	*OBJEXIST, *USE	*EXECUTE
	Message queue [5]	*OBJEXIST, *USE, *DLT	*EXECUTE
DSPACTPRFL [14](Q)			
DSPACTSCD [14](Q)			
DSPAUTUSR [6]	User profile	*READ	
DSPEXPSCD [14](Q)			
DSPPGMADP	User profile	*OBJMGT	
	Output file	See General Rules on page 311	See General Rules on page 311
DSPUSRPRF	User profile	*READ	*EXECUTE
	Output file	See General Rules on page 311	See General Rules on page 311
DSPUSRPRTI	User profile	*USE	
GRTUSRAUT [7]	Referenced user profile	*READ	
	Objects you are granting authority to	*OBJMGT	*EXECUTE
PRTPRFINT [14](Q)			
PRTUSRPRF [14](Q)			
RSTAUT (Q) [8]			

User Profile Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
RSTUSRPRF (Q) [8,10,16]			
RTVUSRPRF	User profile	*READ	
RTVUSRPRTI	User profile	*USE	
SAVSECDTA [8]	Save file, if empty	*USE, *ADD	*EXECUTE
	Save file, if records exist	*OBJMGT, *USE, *ADD	*EXECUTE
WRKUSRPRF [13]	User profile	Any authority	

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 *OBJMGT 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 "Commands Common for Most Objects" on page 313.

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 "Commands Common for Most Objects" on page 313.

10 You must have *ALLOBJ special authority to specify ALWOBJDIF(*ALL).

11 You must have *AUDIT special authority.

12 The user whose profile is created is given these authorities to it: *OBJMGT, *OBJOPR, *READ, *ADD, *DLT, *UPD, *EXECUTE.

13 To use an individual operation, you must have the authority required by the operation.

14 You must have *ALLOBJ special authority to use this command.

15 You must have *JOBCTL special authority to use this command.

16 You must have *ALLOBJ and *SECADM special authorities to specify SECDTA(*PWDGRP), USRPRF(*ALL) or OMITUSRPRF.

17 When you perform a CRTUSRPRF, you can not create a user profile (*USRPRF) into an independent disk pool. However, when a user is privately authorized to an object in the independent disk pool, is the owner of an object on an independent disk pool, or is the primary primary group of an object on an independent disk pool, the name of the profile is stored on the independent disk pool. If the independent disk pool is moved to another system, the private authority, object ownership, and primary group entries will be attached to the profile with the same name on the target system. If a profile does not exist on the target system, a profile will be created. The user will not have any special authorities and the password will be set to *NONE.

User-Defined File System

Command	Referenced Object	Object Type	File System	Authority Needed for Object
ADDMFS [1,2,3,4]	/dev/QASPxx	*DIR	"root"	*W, *RX
	/dev/QASPxx/yyy	*BLKSF[8]	"root"	*R
	dir_to_be_ mounted_over	*DIR	"root"	*W
CRTUDFS [1,2] (Q)	/dev/QASPxx	*DIR	"root"	*RWX
DLTUDFS [1,2,6,7] (Q)	/dev/QASPxx	*DIR	"root"	*RWX
	any_epfs_object		"root"	*RWX, *OBJEXIST
DSPUDFS	some_dirsxx	*DIR	"root"	*RX
MOUNT [1,2,3,4]	/dev/QASPxx	*DIR	"root"	*RWX
	/dev/QASPxx/yyy	*BLKSF	"root"	*R
	dir_to_be_ mounted_over	*DIR	"root"	*W
RMVMFS [1,4]	some_dirs	*DIR	"root"	n/a
UNMOUNT [1,4]	some_dirs	*DIR	"root"	n/a

[1] To use this command, you must have *IOSYSCFG special authority.

[2] QASPxx is either 01 (system asp) or 02-16 based on which user asp is needed. This is the directory that contains the *BLKSF that is being mounted.

[3] The directory that is mounted over (dir_to_be_mounted_over) is any IFS directory that can be mounted over.

[4] You must provide a path to some object. You must have *X authority for all directories in that path.

[5] You must have *RX authority to the /etc/exports stream file and the directories in the /etc/exports path.

[6] A UDFS can contain an entire subtree of EPFS objects, so when you delete a UDFS, you delete objects of all types that can be stored in an EPFS file system.

[7] When using the DLTUDFS commands, you must have *OBJEXIST authority on every object in the UDFS or no objects are deleted.

[8] This object is allowed on independent disk pools.

Validation List Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTVLDL	Validation list		*ADD, *READ
DLTVLDL	Validation list	*OBJEXIST	*EXECUTE

Workstation Customizing Commands

Command	Referenced Object	Authority Needed	
		For Object	For Library
CRTWSCST	Source file	*USE	*EXECUTE
	Workstation customizing object, if REPLACE(*NO)		*READ, *ADD
	Workstation customizing object, if REPLACE(*YES)	*OBJMGT, *OBJEXIST	*READ, *ADD
DLTWSCST	Workstation customizing object	*OBJEXIST	*EXECUTE
RTVWSCST	To-file, if it exists and a new member is added	*OBJOPR, *OBJMGT, *ADD	*EXECUTE
	To-file, if file and member exist	*OBJOPR, *ADD, *DLT	*EXECUTE
	To-file, if the file does not exist		*READ, *ADD

Writer Commands

Command	Referenced Object	Authority Needed		Output Queue Parameters		Special Authority
		For Object	For Library	AUTCHK	OPRCTL	
CHGWTR [2, 4]	Current output queue[1]	*READ, *ADD, *DLT	*EXECUTE	*DTAAUT		
		Owner [3]	*EXECUTE	*OWNER		
					*YES	*JOBCTL
ENDWTR [1]	Output queue	*READ, *ADD, *DLT	*EXECUTE	*DTAAUT		
		Owner [3]	*EXECUTE	*OWNER		
					*YES	*JOBCTL
HLDWTR [1]	Output queue	*READ, *ADD, *DLT	*EXECUTE	*DTAAUT		
		Owner [3]	*EXECUTE	*OWNER		
					*YES	*JOBCTL
RLSWTR [1]	Output queue	*READ, *ADD, *DLT	*EXECUTE	*DTAAUT		
		Owner [3]	*EXECUTE	*OWNER		
					*YES	*JOBCTL
STRDKTWTR [1]	Output queue	*READ, *ADD, *DLT	*EXECUTE	*DTAAUT		
		Owner [3]	*EXECUTE	*OWNER		
			*EXECUTE		*YES	*JOBCTL
	Message queue	*OBJOPR, *ADD	*EXECUTE			
	Device description	*OBJOPR, *READ				

Command	Referenced Object	Authority Needed		Output Queue Parameters		Special Authority
		For Object	For Library	AUTCHK	OPRCTL	
STRPRTWTR [1]	Output queue	*READ, *ADD, *DLT	*EXECUTE	*DTAAUT		
		Owner [3]	*EXECUTE	*OWNER		
			*EXECUTE		*YES	*JOBCTL
	Message queue	*OBJOPR, *ADD	*EXECUTE			
	User-defined driver program	*READ	*EXECUTE			
	Data transform program	*READ	*EXECUTE			
	Separator program	*READ	*EXECUTE			
	Device description	*OBJOPR, *READ				
STRRMTWTR [1]	Output queue	*READ, *ADD, *DLT	*EXECUTE	*DTAAUT		
		Owner [3]	*EXECUTE	*OWNER		
			*EXECUTE		*YES	*JOBCTL
		Message queue	*OBJOPR, *ADD	*EXECUTE		
		User driver program	*READ	*EXECUTE		
	User data transform	*READ	*EXECUTE			
WRKWTR						

[1] If you have *SPLCTL special authority, you do not 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.

[4] You must have *EXECUTE authority to the new output queue's library even if the user has *SPLCTL special authority.

Appendix E. 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 263 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

  SAV Save Object in Directory

  SAVCHGOBJ
  Save Changed Object

  SAVLIB
  Save Library

  SAVOBJ
  Save Object

  SAVSAVFDTA
  Save Save File Data

  SAVDLO
  Save DLO Object

  SAVLICPGM
  Save Licensed Program

  SAVSHF
  Save Bookshelf

  Note: The audit record for the save operation will identify if the save was done with the STG(*FREE).

- Change operation

  APYJRNCHG
  Apply Journaled Changes

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.

ENDJRNxxx
End Journaling

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

QjoEndJournal
End Journaling

QjoStartJournal
Start Journaling

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.

RMVJRNCHG
Remove Journaled Changes

RNMOBJ
Rename Object

RST Restore Object in Directory

RSTCFG
Restore Configuration Objects

RSTLIB
Restore Library

RSTLICPGM
Restore Licensed Program

RSTOBJ
Restore Object

RVKOBJAUT
Revoke Object Authority

STRJRNxxx
Start Journaling

- Operations that are not audited

Prompt [3]
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.

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

3. 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.

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 Access Path Recovery Times:

Note: Changes to access path recovery times 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

  CHGRCYAP
  Change Recovery for Access Paths

  EDTRCYAP
  Edit Recovery for Access Paths
- Operations that are not audited

  DSPRCYAP
  Display Recovery for Access Paths

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

  RTVBNDSRC
  Retrieve Binder Source

  UPDPGM
  Update Program

UPDSRVPGM
Update 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 Special Files (*CHRSF):

See Operations for Stream File (*STMF) for *CHRSF auditing.

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 Change Request Description (*CRQD):

- Read operation

  QFVLSTA
  List Change Request Description Activities API

  QFVRTVCD
  Retrieve Change Request Description API

  SBMCRQ
  Submit Change Request
- Change operation

  ADDCMDCRQA
  Add Command Change Request Activity

  ADDOBJCRQA
  Add Object Change Request Activity

  ADDPRDCRQA
  Add Product Change Request Activity

  ADDPTFCRQA
  Add PTF Change Request Activity

  ADDRSCCRQA
  Add Resource Change Request Activity

  CHGCMDCRQA
  Change Command Change Request Activity

  CHGCRQD
  Change Change Request Description

  CHGOBJCRQA
  Change Object Change Request Activity

  CHGPRDCRQA
  Change Product Change Request Activity

  CHGPTFCRQA
  Change PTF Change Request Activity

  CHGRSCCRQA
  Change Resource Change Request Activity

  QFVADDA
  Add Change Request Description Activity API

  QFVRMVA
  Remove Change Request Description Activity API

  RMVCRQDA
  Remove Change Request Dscription Activity
- Operations that are not audited

  WRKCRQD
  Work with Change Request Descriptions

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]	ENDPGM	RCVF
CALLPRC	ENDRCV	RETURN
CHGVAR	GOTO	SNDF
COPYRIGHT	IF	SNDRCVF
DCL	MONMSG	TFRCTL
DCLF	PGM	WAIT
DO		
ELSE		
ENDDO		

[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

SAVCFG
Save Configuration

VFYCMN
Link test

- Change operation

CHGCTLxxx
Change controller description

VRYCFG
Vary controller description on or off

- Operations that are not audited

DSPCTLD
Display Controller Description

ENDCTLRCY
End Controller Recovery

PRTDEVADR
Print Device Address

RSMCTLRCY
Resume Controller Recovery

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 or explicit acquire operation

Allocate
Allocate conversation

SAVCFG
Save Configuration

STRPASTHR
Start pass-through session

Start of the second session for intermediate pass-through

VFYCMN
Link test

- Change operation

CHGDEVxxx
Change device description

HLDDEVxxx
Hold device description

RLSDEVxxx
Release device description

QWSSETWS
Change type-ahead setting for a device

VRYCFG
Vary device description on or off

- Operations that are not audited

DSPDEVD
Display Device Description

DSPMODSTS
Display Mode Status

ENDDEVRCY
End Device Recovery

HLDCMNDEV
Hold Communications Device

RLSCMNDEV
Release Communications Device

RSMDEVRCY
Resume Device Recovery

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 (*DIR):

- Read/search operations

access, accessx, QlgAccess, QlgAccessx
Determine file accessibility

CHGATR
Change Attribute

CPY Copy Object

DSPCURDIR
Display Current Directory

DSPLNK
Display Links

faccessx
Determine file accessibility for a class of users by descriptor

getcwd, qlgGetcwd
Get Path Name of Current Directory API

givedescriptor
Give File Access API

Qp0lGetAttr, QlgGetAttr
Get attributes APIs

Qp0lGetPathFromFileID, QlgGetPathFromFileID
Get Path From File APIs

Qp0lProcessSubtree, QlgProcessSubtree
Process a Path Name APIs

open, open64, QlgOpen, QlgOpen64, Qp0lOpen
Open File APIs

Qp0lSetAttr, QlgSetAttr
Set Attributes APIs

opendir, QlgOpendir
Open Directory APIs

RTVCURDIR
Retrieve Current Directory

SAV Save

WRKLNK
Work with Links

- Change operation

CHGATR
Change Attributes

CHGAUD
Change Auditing

CHGAUT
Change Authority

CHGOWN
Change Owner

CHGPGP
Change Primary Group

chmod, QlgChmod
Change File Authorizations API

chown, QlgChown
Change Owner and Group API

CPY Copy

CRTDIR
Create Directory

fchmod
Change File Authorizations by Descriptor API

fchown
Change Owner and Group of File by Descriptor API

givedescriptor
Give File Access API

mkdir, QlgMkdir
Make Directory API

MOV Move

Qp0lRenameKeep, QlgRenameKeep
Rename File or Directory, Keep New APIs

Qp0lRenameUnlink, QlgRenameUnlink
Rename File or Directory, Unlink New APIs

Qp0lSetAttr, QlgSetAttr
Set Attribute APIs

rmdir, QlgRmdir
Remove Directory API

RMVDIR
Remove Directory

RNM Rename

RST Restore

utime, QlgUtime
Set File Access and Modifcation Times API

WRKAUT
Work with Authority

WRKLNK
Work with Links

- Operations that are not audited

chdir, QlgChdir
Change Directory API

CHGCURDIR
Change Current Directory

close Close File Descriptor API

closedir
Close Directory API

DSPAUT
Display Authority

dup Duplicate Open File Descriptor API

dup2 Duplicate Open File Descriptor to Another Descriptor API

faccessx
Determine file accessibility for a class of users by descriptor

fchdir Change current directory by descriptor

fcntl Perform File Control Command API

fpathconf
Get Configurable Path Name Variables by Descriptor API

fstat, fstat64
Get File Information by Descriptor APIs

givedescriptor
Give File Access API

ioctl Perform I/O Control Request API

lseek, lseek64
Set File Read/Write Offset APIs

lstat, lstat64, QlgLstat, QlgLstat64
Get File or Link Information APIs

pathconf, QlgPathconf
Get Configurable Path Name Variables API

readdir
Read Directory Entry API

rewinddir
Reset Directory Stream API

select Check I/O Status of Multiple File Descriptors API

stat, QlgStat
Get File Information API

takedescriptor
Take File Access API

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.

CHGSYSDIRA
Change System 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, renaming, 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 WRKDIRE command or using F4 to select entries for sending a note.

Locations
Adding, changing, deleting, or displaying directory location data

Nickname
Adding, changing, renaming 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

SAVSHF
Save bookshelf

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

CHGDLOPGP
Change DLO Primary Group

CHGDOCD
Change Document Description

CHGDSTD
Change Distribution Description

CPYDOC [4]
Copy Document

Note: A change entry is written if the target document already exists.

CRTFLR
Create Folder

CVTTOFLR [4]
Convert to Folder

DLTDLO [4]
Delete DLO

DLTSHF
Delete Bookshelf

DTLDOCL [4]
Delete Document List

DLTDST [4]
Delete Distribution

EDTDLOAUT
Edit DLO Authority

EDTDOC
Edit Document

FILDOC [4]
File Document

GRTACCAUT
Grant Access Code Authority

GRTUSRPMN
Grant User Permission

MOVDOC [4]
Move Document

MRGDOC [4]
Merge Document

PAGDOC
Paginate Document

QHFCHGAT
Change Directory Entry Attributes API

QHFSETSZ
Set Stream File Size API

QHFWRTSF
Write Stream File API

QRYDOCLIB [4]
Query Document Library

Note: A change entry is written if an existing document resulting from a search is replaced.

RCVDST [4]
Receive Distribution

4. A change entry is written for both the document and the folder if the target of the operation is in a folder.

RGZDLO
Reorganize DLO

RMVACC
Remove access code, for any DLO to which the access code is attached

RMVDLOAUT
Remove DLO authority

RNMDLO [4]
Rename DLO

RPLDOC
Replace Document

RSTDLO [4]
Restore DLO

RSTSHF
Restore Bookshelf

RTVDOC
Retrieve Document (check out)

RVKACCAUT
Revoke Access Code Authority

RVKUSRPMN
Revoke User Permission

SAVDLO [4]
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

QMHRDQM
Retrieve Data Queue Message API

- Change operation

QRCVDTAQ
Receive Data Queue API

QSNDDTAQ
Send Data Queue API

QCLRDTAQ
Clear Data Queue API

- Operations that are not audited

WRKDTAQ
Work with Data Queue

QMHQRDQD
Retrieve Data Queue Description API

Operations for Edit Description (*EDTD):

- Read operation

DSPEDTD
Display Edit Description

QECCVTEC
Edit code expansion API (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 Exit Registration (*EXITRG):

- Read operation

QUSRTVEI
Retrieve Exit Information API

QusRetrieveExitInformation
Retrieve Exit Information API

- Change operation

ADDEXITPGM
Add Exit Program

QUSADDEP
Add Exit Program API

QusAddExitProgram
Add Exit Program API

QUSDRGPT
Deregister Exit Point API

QusDeregisterExitPoint
Deregister Exit Point API

QUSRGPT
Register Exit Point API

QusRegisterExitPoint
Register Exit Point API

QUSRMVEP
Remove Exit Program API

QusRemoveExitProgram
Remove Exit Program API

RMVEXITPGM
Remove Exit Program

WRKREGINF
Work with Registration Information

- Operations that are not audited

  None

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

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

ADDPFCST
Add Physical File Constraint

ADDPFM
Add Physical File Member

ADDPFTRG
Add Physical File Trigger

ADDPFVLM
Add Physical File Variable Length Member

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

CHGPFCST
Change Physical File Constraint

CHGPFM
Change Physical File Member

CHGPRTF
Change Printer Device GQle

CHGSAVF
Change Save File

CHGS36PRCA
Change S/36 Procedure Attributes

CHGS36SRCA
Change S/36 Source Attributes

CHGTAPF
Change Tape Device File

CLRPFM
Clear Physical File Member

CPYF Copy File (open file for modification, such as adding records, clearing a member, or saving a member

EDTS36PRCA
Edit S/36 Procedure Attributes

EDTS36SRCA
Edit S/36 Source Attributes

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

RMVM
Remove Member

RMVPFCST
Remove Physical File Constraint

RMVPFTGR
Remove Physical File Trigger

RNMM
Rename Member

WRKS36PRCA
Work with S/36 Procedure Attributes

WRKS36SRCA
Work with S/36 Source Attributes

- Operations that are not audited

DSPCPCST
Display Check Pending Constraints

DSPFD
Display File Description

DSPFFD
Display File Field Description

DSPDBR
Display Database Relations

DSPPGMREF
Display Program File References

EDTCPCST
Edit Check Pending Constraints

OVRxxx
Override file

RTVMBRD
Retrieve Member Description

WRKPFCST
Work with Physical File Constraints

WRKF
Work with File

Operations for First-in First-out Files (*FIFO):

- See Operations for Stream File (*STMF) for the *FIFO auditing.

Operations for Folder (*FLR):

- See operations for Document Library Object (*DOC or *FLR)

Operations for Font Resource (*FNTRSC):

- Read operation

  Print Printing a spooled file that refers to the font resource

- Change operation

  None

- Operations that are not audited

  WRKFNTRSC
  Work with Font Resource

  Print Referring to the font resource when creating a spooled file

Operations for Form Definition (*FORMDF):

- Read operation

  Print Printing a spooled file that refers to the form definition

- Change operation

  None

- Operations that are not audited

  WRKFORMDF
  Work with Form Definition

  Print Referring to the form definition when creating a spooled file

Operations for Filter Object (*FTR):

- Read operation

  None

- Change operation

  ADDALRACNE
  Add Alert Action Entry

  ADDALRSLTE
  Add Alert Selection Entry

  ADDPRBACNE
  Add Problem Action Entry

  ADDPRBSLTE
  Add Problem Selection Entry

  CHGALRACNE
  Change Alert Action Entry

  CHGALRSLTE
  Change Alert Selection Entry

  CHGPRBACNE
  Change Problem Action Entry

  CHGPRBSLTE
  Change Problem 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*)

  Conversion
  Conversion to V3R1 format, if necessary

  Print Print character to register in sort table (option 1 from CGU menu)

  Print before deleting character from sort table (option 2 from CGU menu)
- Change operation

  CPYIGCSRT
  Copy IGC Sort (*to-*ICGSRT-object*)

  Conversion
  Conversion to V3R1 format, if necessary

  Create Create a user-defined character (option 1 from CGU menu)

  Delete Delete a user-defined character (option 2 from CGU menu)

  Update
  Update the active sort table (option 5 from CGU menu)
- Operations that are not audited

  FMTDTA
  Sort records or fields in a file

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 "Subsystem Descriptions" on page 191
  Add Job Queue Entry

---

5. An audit record is written if object auditing is specified for the subsystem description (*SBSD).

CHGJOB
Change Job from one JOBQ to another JOBQ

CHGJOBQE "Subsystem Descriptions" on page 191
Change Job Queue Entry

QSPRJOBQ
Retrieve job queue information

RMVJOBQE "Subsystem Descriptions" on page 191
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 for user journals

QJORJIDI
Retrieve Journal Identifier (JID) Information

QjoRetrieveJournalEntries
Retrieve Journal Entries

RCVJRNE
Receive Journal Entry

RTVJRNE
Retrieve Journal Entry

- Change operation

ADDRMTJRN
Add Remote Journal

APYJRNCHG
Apply Journaled Changes

CHGJRN
Change Journal

CHGRMTJRN
Change Remote Journal

ENDJRN*xxx*
End Journaling

JRNAP
(S/38E) Start Journal Access Path

JRNPF
(S/38E) Start Journal Physical File

QjoAddRemoteJournal
Add Remote Journal API

QjoChangeJournalState
Change Journal State API

QjoEndJournal
End Journaling API

QjoRemoveRemoteJournal
Remove Remote Journal API

QJOSJRNE
Send Journal Entry API (user entries only via QJOSJRNE API)

QjoStartJournal
Start Journaling API

RMVJRNCHG
Remove Journaled Changes

RMVRMTJRN
Remove Remote Journal

SNDJRNE
Send Journal Entry (user entries only via SNDJRNE command)

STRJRN*xxx*
Start Journaling

- Operations that are not audited

  DSPJRN
  Display Journal Entry for internal system journals, JRN(*INTSYSJRN)

  DSPJRNA
  (S/38E) Work with Journal Attributes

  DSPJRNMNU
  (S/38E) Work with Journal

  QjoRetrieveJournalInformation
  Retrieve Journal Information API

  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

  QjoRtvJrnReceiverInformation
  Retrieve Journal Receiver Information API

  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*/*ALL) +
     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

Operations for Line Description (*LIND):

- Read operation

  SAVCFG
  Save Configuration

  RUNLPDA
  Run LPDA-2 operational commands

  VFYCMN
  Link test

  VFYLNKLPDA
  LPDA-2 link test
- Change operation

  CHGLIN*xxx*
  Change Line Description

  VRYCFG
  Vary on/off line description
- Operations that are not audited

  ANSLIN
  Answer Line

  Copy Option 3 from WRKLIND

  DSPLIND
  Display Line Description

  ENDLINRCY
  End Line Recovery

  RLSCMNDEV
  Release Communications Device

  RSMLINRCY
  Resume Line Recovery

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

CHGSSNMAX
Change session maximum

DSPMODD
Display Mode Description

ENDMOD
End Mode

STRMOD
Start Mode

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

UPDPGM
An audit entry for each module object used during an UPDPGM

UPDSRVPGM
An audit entry for each module object used during an UPDSRVPGM

- Change operation

CHGMOD
Change Module

- Operations that are not audited

DSPMOD
Display Module

RTVBNDSRC
Retrieve Binder Source

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

  QMHRTVM
  Retrieve Message API

  WRKMSGD
  Work with Message Description

- Change operation

  ADDMSGD
  Add Message Description

  CHGMSGD
  Change Message Description

  CHGMSGF
  Change Message File

  MRGMSGF
  Merge Message File (to-file and replace MSGF)

  RMVMSGD
  Remove Message Description

- Operations that are not audited

  OVRMSGF
  Override Message File

  WRKMSGF
  Work with Message File

  QMHRMFAT
  Retrieve Message File Attributes API

Operations for Message Queue (*MSGQ):

- Read operation

  QMHLSTM
  List Nonprogram Messages API

  QMHRMQAT
  Retrieve Nonprogram Message Queue Attributes API

DSPLOG
Display Log

DSPMSG
Display Message

Print Print Messages

RCVMSG
Receive Message RMV(*NO)

QMHRCVM
Receive Nonprogram Messages API when message action is not *REMOVE.

- Change operation

CHGMSGQ
Change Message Queue

CLRMSGQ
Clear Message Queue

RCVMSG
Receive Message RMV(*YES)

QMHRCVM
Receive Nonprogram Messages API when message action is *REMOVE.

RMVMSG
Remove Message

QMHRMVM
Remove Nonprogram Messages API

SNDxxxMSG
Send a Message to a message queue

QMHSNDBM
Send Break Message API

QMHSNDM
Send Nonprogram Message API

QMHSNDRM
Send Reply Message API

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 Group (*NODGRP):

- Read operation

DSPNODGRP
Display Node Group

- Change operation

  CHGNODGRPA
  Change Node Group

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 NetBIOS Description (*NTBD):

- Read operation

  SAVCFG
  Save Configuration

- Change operation

  CHGNTBD
  Change NetBIOS Description

- Operations that are not audited

  Copy Option 3 of WRKNTBD

  DSPNTBD
  Display NetBIOS Description

  RTVCFGSRC
  Retrieve Configuration Source of NetBIOS description

  WRKNTBD
  Work with NetBIOS Description

Operations for Network Interface (*NWID):

- Read operation

  SAVCFG
  Save Configuration

- Change operation

  CHGNWIISDN
  Change Network Interface Description

  VRYCFG
  Vary network interface description on or off

- Operations that are not audited

  Copy Option 3 of WRKNWID

DSPNWID
Display Network Interface Description

ENDNWIRCY
End Network Interface Recovery

RSMNWIRCY
Resume Network Interface Recovery

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 Network Server Description (*NWSD):

- Read operation

  SAVCFG
  Save Configuration

- Change operation

  CHGNWSD
  Change Network Server Description

  VRYCFG
  Vary Configuration

- Operations that are not audited

  Copy Option 3 of WRKNWSD

  DSPNWSD
  Display Network Server Description

  RTVCFGSRC
  Retrieve Configuration Source for *NWSD

  RTVCFGSTS
  Retrieve Configuration Status for *NWSD

  WRKNWSD
  Work with Network Server Description

Operations for Output Queue (*OUTQ):

- Read operation

  STRPRTWTR
  Start a Printer Writer to an OUTQ

  STRRMTWTR
  Start a Remote Writer to an OUTQ

- Change operation

  Placement
  When an entry is placed on or removed from the queue

  CHGOUTQ
  Change Output Queue

CHGSPLFA [6]
Change Spooled File Attributes, if moved to a different output queue and either output queue is audited

CLROUTQ
Clear Output Queue

DLTSPLF [6]
Delete Spooled File

HLDOUTQ
Hold Output Queue

RLSOUTQ
Release Output Queue

- Operations that are not audited

CHGSPLFA [6]
Change Spooled File Attributes

CPYSPLF [6]
Copy Spooled File

Create [6]
Create a spooled file

DSPSPLF [6]
Display Spooled File

HLDSPLF [6]
Hold Spooled File

QSPROUTQ
Retrieve output queue information

RLSSPLF [6]
Release Spooled File

SNDNETSPLF [6]
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 Printing a spooled file that refers to the overlay

- Change operation

  None

- Operations that are not audited

6. This is also audited if action auditing (QAUDLVL system value or AUDLVL user profile value) includes *SPLFDTA.

WRKOVL
Work with overlay

Print Referring to the overlay when creating a spooled file

Operations for Page Definition (*PAGDFN):

- Read operation

  Print Printing a spooled file that refers to the page definition
- Change operation

  None
- Operations that are not audited

  WRKPAGDFN
  Work with Page Definition

  Print Referring to the form definition when creating a spooled file

Operations for Page Segment (*PAGSEG):

- Read operation

  Print Printing a spooled file that refers to the page segment
- Change operation

  None
- Operations that are not audited

  WRKPAGSEG
  Work with Page Segment

  Print Referring to the page segment when creating a spooled file

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

- Create operation

CRTPGM
Create Program

UPDPGM
Update Program

- 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

PRTSQLINF
Print SQL Information

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 Query Manager Forms function

- Operations that are not audited

  Work with
  When *QMFORMs are listed in a Work with display

  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

  CRTQMQRY
  Create Query Management Query

  Convert
  Option 10 (Convert to SQL) from the Work with Query Manager Queries function

Copy Option 3 from the Work with Query Manager Queries function

Save Save the query using a menu or command

- 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 or QRY.

  Display
  Display a query using WRKQRY prompt display

  Export Export form using Query Manager

  Export Export query using Query Manager

  Print Print query definition using WRKQRY prompt display

  Print Query Management form

  Print Query Management query

  Print Query Management report

  QRYRUN
  Run Query

  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

  Copy Copy a query using option 3 on the "Work with Queries" display

  Create Create a query using option 1 on the "Work with Queries" display

  Delete Delete a query using option 4 on the "Work with Queries" display

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

DLTQRY
Delete a query

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 Local Socket (*SOCKET):

- Read operation

  connect
  Bind a permanent destination to a socket and establish a connection.

  DSPLNK
  Display Links

  givedescriptor
  Give File Access API

  Qp0lGetPathFromFileID
  Get Path Name of Object from File ID API

  Qp0lRenameKeep
  Rename File or Directory, Keep New API

  Qp0lRenameUnlink
  Rename File or Directory, Unlink New API

  sendmsg
  Send a datagram in connectionless mode. Can use multiple buffers.

  sendto
  Send a datagram in connectionless mode.

  WRKLNK
  Work with Links

- Change operation

  ADDLNK
  Add Link

  bind Establish a local address for a socket.

  CHGAUD
  Change Auditing

  CHGAUT
  Change Authority

CHGOWN
Change Owner

CHGPGP
Change Primary Group

CHKIN
Check In

CHKOUT
Check Out

chmod
Change File Authorizations API

chown
Change Owner and Group API

givedescriptor
Give File Access API

link Create Link to File API

Qp0lRenameKeep
Rename File or Directory, Keep New API

Qp0lRenameUnlink
Rename File or Directory, Unlink New API

RMVLNK
Remove Link

RNM Rename

RST Restore

unlink
Remove Link to File API

utime Set File Access and Modifcation Times API

WRKAUT
Work with Authority

WRKLNK
Work with Links

- Operations that are not audited:

  close Close File API

  DSPAUT
  Display Authority

  dup Duplicate Open File Descriptor API

  dup2 Duplicate Open File Descriptor to Another Descriptor API

  fcntl Perform File Control Command API

  fstat Get File Information by Descriptor API

  fsync Synchronize Changes to File API

  ioctl Perform I/O Control Request API

  lstat Get File or Link Information API

  pathconf
  Get Configurable Path Name Variables API

read Read from File API

readv Read from File (Vector) API

select Check I/O Status of Multiple File Descriptors API

stat Get File Information API

takedescriptor
Take File Access API

write Write to File API

writev Write to File (Vector) API

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
  - STRRMTWTR

– 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))
– Sending a spooled file to a remote system by a remote writer

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 SAVE(*YES)
– Sending a spooled file to a remote system by a remote writer when the spooled file specifies SAVE(*YES)
– Having a writer hold a spooled file after an error occurs when processing the spooled file

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

  PRTSQLINF
  Print SQL Information

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

RTVBNDSRC
Retrieve Binder Source

UPDPGM
An audit entry for each service program used during a UPDPGM command.

UPDSRVPGM
An audit entry for each service program used during a UPDSRVPGM command.

- Create operation

  CRTSRVPGM
  Create Service Program

  UPDSRVPGM
  Update Service Program

- Change operation

  CHGSRVPGM
  Change Service Program

- Operations that are not audited

  DSPSRVPGM
  Display Service Program

  PRTSQLINF
  Print SQL Information

  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 Server Storage Space (*SVRSTG):

- No Read or Change operations are audited for the *SVRSTG object type.

Operations for Stream File (*STMF):

- Read operation

  CPY Copy

  DSPLNK
  Display Links

givedescriptor
Give File Access API

MOV Move

open, open64, QlgOpen, QlgOpen64, Qp0lOpen
Open File APIs

SAV Save

WRKLNK
Work with Links

- Change operation

ADDLNK
Add Link

CHGAUD
Change Auditing

CHGAUT
Change Authority

CHGOWN
Change Owner

CHGPGP
Change Primary Group

CHKIN
Check In

CHKOUT
Check Out

chmod, QlgChmod
Change File Authorizations APIs

chown, QlgChown
Change Owner and Group APIs

CPY Copy

creat, creat64, QlgCreat, QlgCreat64
Create New File or Rewrite Existing File APIs

fchmod
Change File Authorizations by Descriptor API

fchown
Change Owner and Group of File by Descriptor API

givedescriptor
Give File Access API

link Create Link to File API

MOV Move

open, open64, QlgOpen, QlgOpen64, Qp0lOpen
When opened for write APIs

Qp0lGetPathFromFileID, QlgGetPathFromFileID
Get Path Name of Object from File ID APIs

Qp0lRenameKeep, QlgRenameKeep
Rename File or Directory, Keep New APIs

Qp0lRenameUnlink, QlgRenameUnlink
Rename File or Directory, Unlink New APIs

RMVLNK
Remove Link

RNM Rename

RST Restore

unlink, QlgUnlink
Remove Link to File APIs

utime, QlgUtime
Set File Access and Modifcation Times APIs

WRKAUT
Work with Authority

WRKLNK
Work with Links

- Operations that are not audited

close Close File API

DSPAUT
Display Authority

dup Duplicate Open File Descriptor API

dup2 Duplicate Open File Descriptor to Another Descriptor API

faccessx
Determine file accessibility

fcntl Perform File Control Command API

fpathconf
Get Configurable Path Name Variables by Descriptor API

fstat, fstat64
Get File Information by Descriptor APIs

fsync Synchronize Changes to File API

ftruncate, ftruncate64
Truncate File APIs

ioctl Perform I/O Control Request API

lseek, lseek64
Set File Read/Write Offset APIs

lstat, lstat64
Get File or Link Information APIs

pathconf, QlgPathconf
Get Configurable Path Name Variables APIs

pread, pread64
Read from Descriptor with Offset APIs

pwrite, pwrite64
Write to Descriptor with Offset APIs

read Read from File API

readv Read from File (Vector) API

select Check I/O Status of Multiple File Descriptors API

stat, stat64, QlgStat, QlgStat64
Get File Information APIs

takedescriptor
Take File Access API

write Write to File API

writev Write to File (Vector) API

Operations for Symbolic Link (*SYMLNK):

- Read operation

  CPY Copy

  DSPLNK
  Display Links

  MOV Move

  readlink
  Read Value of Symbolic Link API

  SAV Save

  WRKLNK
  Work with Links

- Change operation

  CHGOWN
  Change Owner

  CHGPGP
  Change Primary Group

  CPY Copy

  MOV Move

  Qp0lRenameKeep, QlgRenameKeep
  Rename File or Directory, Keep New APIs

  Qp0lRenameUnlink, QlgRenameUnlink
  Rename File or Directory, Unlink New APIs

  RMVLNK
  Remove Link

  RNM Rename

  RST Restore

  symlink, QlgSymlink
  Make Symbolic Link APIs

  unlink, QlgUnlink
  Remove Link to File APIs

  WRKLNK
  Work with Links

- Operations that are not audited

  lstat, lstat64, QlgLstat, QlgLstat64
  Link Status APIs

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 user 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

CHKPWD
Check Password

DLTUSRPRF
Delete User Profile

GRTUSRAUT
Grant User Authority (*to-user-profile*)

QSYCHGPW
Change Password API

RSTUSRPRF
Restore User Profile

- Operations that are not audited

DSPPGMADP
Display Programs that Adopt

DSPUSRPRF
Display User Profile

GRTUSRAUT
Grant User Authority (*from-user-profile*)

PRTPRFINT
Print Profile Internals

PRTUSRPRF
Print 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 that are not audited

  Access
  Direct access to user queues using MI instructions (only allowed for a user domain user queue in a library specified in the QALWUSRDMN system value.

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 user spaces in libraries specified in the QALWUSRDMN system value.

  QUSRUSAT
  Retrieve User Space Attributes API

Operations for Validation List (*VLDL):

- Read operation

  QSYFDVLE
  Find Validation List Entry API
- Change operation

  QSYADVLE
  Add Validation List Entry API

QSYCHVLE
Change Validation List Entry API

QSYRMVLE
Remove Validation List Entry API

- Operations that are not audited

Access
Direct access to user space using MI instructions (only allowed for user domain user spaces 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)

SNDTCPSPLF
Send TCP/IP Spooled File (only when TRANSFORM(*YES) is specified)

STRPRTWTR
Start Printer Writer (only for spooled files that are printed to a customized printer using the host print transform function)

STRRMTWTR
Start Remote Writer (only when output queue is configured with CNNTYPE(*IP) and TRANSFORM(*YES))

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

Appendix F. Layout of Audit Journal Entries

This appendix contains layout information for all entry types with journal code T in the audit (QAUDJRN) journal. These entries are controlled by the action and object auditing you define. The system writes additional entries to the audit journal for such events as a system IPL or saving the journal receiver. The layouts for these entry types can be found in the *Backup and Recovery* book.

Table 143 on page 504 contains the layout for fields that are common to all entry types when OUTFILFMT(*TYPE2) is specified on the DSPJRN command. This layout, which is called QJORDJE2, is defined in the QADSPJR2 file in the QSYS library.

Note: TYPE2 and *TYPE 4 output formats are no longer updated; therefore, IBM recommends that you stop using *TYPE2 and *TYPE4 formats and use only *TYPE5 formats.

Table 142 on page 503 contains the layout for fields that are common to all entry types when OUTFILFMT(*TYPE4) is specified on the DSPJRN command. This layout, which is called QJORDJE4, is defined in the QADSPJR4 file in the QSYS library. The *TYPE4 output includes all of the *TYPE2 information, plus information about journal identifiers, triggers, and referential constraints.

Table 145 on page 506 through Table 217 on page 596 contain layouts for the model database outfiles 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 model database outfiles. 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 274 provides examples of using the model database outfiles. See also the *Journal Entry Information* Appendix in the *Backup and Recovery* book for detailed descriptions for these fields.

Table 141 contains the layout for fields that are common to all entry types when OUTFILFMT(*TYPE5) is specified on the DSPJRN command. This layout, which is called QJORDJE5, is defined in the QADSPJR5 file in the QSYS library. The *TYPE5 output includes all of the *TYPE4 information, plus information about the program library, program ASP device name, program ASP device number, receiver, receiver library, receiver ASP device name, receiver ASP device number, arm number, thread id, address family, remote port, and remote address.

Table 141. Standard Heading Fields for Audit Journal Entries. QJORDJE5 Record Format (*TYPE5)

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	Char(20)	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.
26	Journal Code	Char(1)	Always T.
27	Entry Type	Char(2)	See Table 144 on page 504 for a list of entry types and descriptions.
29	Timestamp of Entry	Char(26)	Date and time that the entry was made in SAA® timestamp format.
55	Name of Job	Char(10)	The name of the job that caused the entry to be generated.
65	User Name	Char(10)	The user profile name associated with the job[1].
75	Job Number	Zoned(6,0)	The job number.

Table 141. Standard Heading Fields for Audit Journal Entries (continued). QJORDJE5 Record Format (*TYPE5)

Offset	Field	Format	Description
81	Program Name	Char(10)	The name of the program that made the journal entry. This can also be the name of a service program or the partial name of a class file used in a compiled Java program. If an application program or CL program did not cause the entry, the field contains the name of a system-supplied program such as QCMD. The field has the value *NONE if one of the following is true: • The program name does not apply to this entry type. • The program name was not available.
91	Program library	Char(10)	Name of the library that contains the program that added the journal entry.
101	Program ASP device	Char(10)	Name of ASP device that contains the program that added the journal entry.
111	Program ASP number	Zoned(5,0)	Number of the ASP that contains the program that added the journal entry.
116	Name of object	Char(10)	Used for journaled objects. Not used for audit journal entries.
126	Objects Library	Char(10)	Used for journaled objects. Not used for audit journal entries.
136	Member Name	Char(10)	Used for journaled objects. Not used for audit journal entries.
146	Count/RRN	Char(20)	Used for journaled objects. Not used for audit journal entries.
166	Flag	Char(1)	Used for journaled objects. Not used for audit journal entries.
167	Commit Cycle identifier	Char(20)	Used for journaled objects. Not used for audit journal entries.
187	User Profile	Char(10)	The name of the current user profile[1].
197	System Name	Char(8)	The name of the system.
205	Journal identifier	Char(10)	Used for file journaling. Not used for audit journal entries.
215	Referential Constraint	Char(1)	Used for file journaling. Not used for audit journal entries.
216	Trigger	Char(1)	Used for file journaling. Not used for audit journal entries.
217	Incomplete Data	Char(1)	Used for file journaling. Not used for audit journal entries.
218	Ignored by APY/RMVJRNCHG	Char(1)	Used for file journaling. Not used for audit journal entries.
219	Minimized ESD	Char(1)	Used for file journaling. Not used for audit journal entries.
220	Object indicator	Char(1)	Used for file journaling. Not used for audit journal entries.
221	System sequence	Char(20)	A number assigned by the system to each journal entry.
241	Receiver	Char(10)	The name of the receiver holding the journal entry.
251	Receiver library	Char(10)	The name of the library containing the receiver holding the journal entry.
261	Receiver ASP device	Char(10)	Name of ASP device that contains the receiver.
271	Receiver ASP number	Zoned(5,0)	Number of the ASP that contains the receiver holding the journal entry.
276	Arm number	Zoned(5,0)	The number of the disk arm that contains the journal entry.
281	Thread identifier	Hex(8)	Identifies the thread within the process that added the journal entry.
289	Thread identifier hex	Char(16)	Displayable hex version of the thread identifier.
305	Address family	Char(1)	The format of the remote address for this journal entry.
306	Remote port	Zoned(5,0)	The port number of the remote address associated with the journal entry.
311	Remote address	Char(46)	The remote address associated with the journal entry.
357	Logical unit of work	Char(39)	Used for file journaling. Not used for audit journal entries.
396	Transaction ID	Char(140)	Used for file journaling. Not used for audit journal entries.
536	Reserved	Char(20)	Used for file journaling. Not used for audit journal entries.
556	Null value indicators	Char(50)	Used for file journaling. Not used for audit journal entries.

Table 141. Standard Heading Fields for Audit Journal Entries (continued). QJORDJE5 Record Format (*TYPE5)

Offset	Field	Format	Description
606	Entry specific data length	Binary(5)	Length of the entry specific data.

Note: The three fields beginning at offset 55 make up the system job name. In most cases, the User name field at offset 65 and the User profile name field at offset 187 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 142. Standard Heading Fields for Audit Journal Entries. QJORDJE4 Record Format (*TYPE4)

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)	See Table 144 on page 504 for a list of entry types and descriptions.
19	Timestamp of Entry	Char(26)	Date and time that the entry was made in SAA timestamp format.
45	Name of Job	Char(10)	The name of the job that caused the entry to be generated.
55	User Name	Char(10)	The user profile name associated with the job[1].
65	Job Number	Zoned(6,0)	The job number.
71	Program Name	Char(10)	The name of the program that made the journal entry. This can also be the name of a service program or the partial name of a class file used in a compiled Java program. If an application program or CL program did not cause the entry, the field contains the name of a system-supplied program such as QCMD. The field has the value *NONE if one of the following is true: • The program name does not apply to this entry type. • The program name was not available.
81	Object Name	Char(10)	Used for journaled objects. Not used for audit journal entries.
91	Library Name	Char(10)	Used for journaled objects. Not used for audit journal entries.
101	Member Name	Char(10)	Used for journaled objects. Not used for audit journal entries.
111	Count/RRN	Zoned(10)	Used for journaled objects. Not used for audit journal entries.
121	Flag	Char(1)	Used for journaled objects. Not used for audit journal entries.
122	Commit Cycle ID	Zoned(10)	Used for journaled objects. Not used for audit journal entries.
132	User Profile	Char(10)	The name of the current user profile[1].
142	System Name	Char(8)	The name of the system.
150	Reserved	Char(10)	Used for file journaling. Not used for audit journal entries.
160	Referential Constraint	Char(1)	Used for file journaling. Not used for audit journal entries.
161	Trigger	Char(1)	Used for file journaling. Not used for audit journal entries.
162	(Reserved Area)	Char(8)	
170	Null Value Indicators	Char(50)	Used for file journaling. Not used for audit journal entries.
220	Entry Specific Data Length	Binary (4)	Length of the entry specific data.

Note: The three fields beginning at offset 45 make up the system job name. In most cases, the User name field at offset 55 and the User profile name field at offset 132 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.

Audit Journal Entries

Table 143. 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)	See Table 144 for a list of entry types and descriptions.
19	Timestamp	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. This can also be the name of a service program or the partial name of a class file used in a compiled Java program. If an application program or CL program did not cause the entry, the field contains the name of a system-supplied program such as QCMD. The field has the value *NONE if one of the following is true: • The program name does not apply to this entry type. • The program name was not available.
67	Object Name	Char(10)	Used for journaled objects. Not used for audit journal entries.
77	Library Name	Char(10)	Used for journaled objects. Not used for audit journal entries.
87	Member Name	Char(10)	Used for journaled objects. Not used for audit journal entries.
97	Count/RRN	Zoned(10)	Used for journaled objects. Not used for audit journal entries.
107	Flag	Char(1)	Used for journaled objects. Not used for audit journal entries.
108	Commit Cycle ID	Zoned(10)	Used for journaled objects. 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 144. Audit Journal (QAUDJRN) Entry Types.

Entry Type	Description
AD	Auditing changes
AF	Authority failure
AP	Obtaining adopted authority
AU	Attribute changes
CA	Authority changes
CD	Command string audit
CO	Create object
CP	User profile changed, created, or restored
CQ	Change of *CRQD object
CU	Cluster Operations
CV	Connection verification
CY	Cryptographic Configuration
DI	Directory Services
DO	Delete object

Table 144. Audit Journal (QAUDJRN) Entry Types. (continued)

Entry Type	Description
DS	DST security password reset
EV	System environment variables
GR	Generic record
GS	Socket description was given to another job
IP	Interprocess Communication
IR	IP Rules Actions
IS	Internet security management
JD	Change to user parameter of a job description
JS	Actions that affect jobs
KF	Key ring file
LD	Link, unlink, or look up directory entry
ML	Office services mail actions
NA	Network attribute changed
ND	APPN directory search filter violation
NE	APPN end point filter violation
OM	Object move or rename
OR	Object restore
OW	Object ownership changed
O1	(Optical Access) Single File or Directory
O2	(Optical Access) Dual File or Directory
O3	(Optical Access) Volume
PA	Program changed to adopt authority
PG	Change of an object's primary group
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
RQ	Restoring a *CRQD object
RU	Restoring user profile authority
RZ	Changing a primary group during restore
SD	Changes to system distribution directory
SE	Subsystem routing entry changed
SF	Actions to spooled files
SG	Asynchronous Signals
SK	Secure sockets connections
SM	System management changes
SO	Server security user information actions
ST	Use of service tools
SV	System value changed
VA	Changing an access control list
VC	Starting or ending a connection
VF	Closing server files
VL	Account limit exceeded
VN	Logging on and off the network
VO	Validation list actions
VP	Network password error
VR	Network resource access
VS	Starting or ending a server session

Table 144. Audit Journal (QAUDJRN) Entry Types. (continued)

Entry Type	Description
VU	Changing a network profile
VV	Changing service status
X0	Network Authentication
YC	DLO object accessed (change)
YR	DLO object accessed (read)
ZC	Object accessed (change)
ZM	SOM method access
ZR	Object accessed (read)

Table 145. AD (Auditing Change) Journal Entries. QASYADJE/J4/J5 Field Description File

Offset					
JE	J4	J5	Field	Format	Description
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	**D** CHGDLOAUD command **O** CHGOBJAUD command **U** CHGUSRAUD command
157	225	611	Object Name	Char(10)	Name of the object for which auditing was changed.
167	235	621	Library Name	Char(10)	Name of the library for the object.
177	245	631	Object Type	Char(8)	The type of object.
185	253	639	Object Audit Value	Char(10)	The new value specified on the CHGOBJAUD command.
195	263	649	CHGUSRAUD *CMD	Char(1)	Y = Audit commands for this user.
196	264	650	CHGUSRAUD *CREATE	Char(1)	Y = Write an audit record when this user creates an object.
197	265	651	CHGUSRAUD *DELETE	Char(1)	Y = Write an audit record when this user deletes an object.
198	266	652	CHGUSRAUD *JOBDTA	Char(1)	Y = Write an audit record when this user changes a job.
199	267	653	CHGUSRAUD *OBJMGT	Char(1)	Y = Write an audit record when this user moves or renames an object.
200	268	654	CHGUSRAUD *OFCSRV	Char(1)	Y = Write an audit record when this user performs office functions.
201	269	655	CHGUSRAUD *PGMADP	Char(1)	Y = Write an audit record when this user obtains authority through adopted authority.
202	270	656	CHGUSRAUD *SAVRST	Char(1)	Y = Write an audit record when this user saves or restores objects.
203	271	657	CHGUSRAUD *SECURITY	Char(1)	Y = Write an audit record when this user performs security-relevant actions.
204	272	658	CHGUSRAUD *SERVICE	Char(1)	Y = Write an audit record when this user performs service functions.
205	273	659	CHGUSRAUD *SPLFDTA	Char(1)	Y = Write an audit record when this user manipulates spooled files.
206	274	660	CHGUSRAUD *SYSMGT	Char(1)	Y = Write an audit record when this user makes system management changes.

Table 145. AD (Auditing Change) Journal Entries (continued). QASYADJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
207	275	661	CHGUSRAUD *OPTICAL	Char (1)	Y = Write an audit record when this user accesses optical devices.
208	276	662	(Reserved Area)	Char(19)	
227	295	681	DLO Name	Char(12)	Name of the DLO object for which auditing was changed.
239	307	693	(Reserved Area)	Char(8)	
247	315	701	Folder Path	Char(63)	Path of the folder.
310			(Reserved Area)	Char(20)	
	378	764	(Reserved Area)	Char(18)	
	396	782	Object Name Length [1]	Binary(4)	The length of the object name.
330	398	784	Object Name CCSID[1]	Binary(5)	The coded character set identifier for the object name.
334	402	788	Object Name Country or Region ID[1]	Char(2)	The Country or Region ID for the object name.
336	404	790	Object Name Language ID[1]	Char(3)	The language ID for the object name.
339	407	793	(Reserved area)	Char(3)	
342	410	796	Parent File ID[1,2]	Char(16)	The file ID of the parent directory.
358	426	812	Object File ID[1,2]	Char(16)	The file ID of the object.
374	442	828	Object Name[1]	Char(512)	The name of the object.
	954	1340	Object File ID	Char(16)	The file ID of the object.
	970	1356	ASP Name[5]	Char(10)	The name of the ASP device.
	980	1366	ASP Number[5]	Char(5)	The number of the ASP device.
	985	1371	Path Name CCSID	Binary(5)	The coded character set identifier for the absolute path name.
	989	1375	Path Name Country or Region ID	Char(2)	The Country or Region ID for the absolute path name.
	991	1377	Path Name Language ID	Char(3)	The language ID for the absolute path name.
	994	1380	Path Name Length	Binary(4)	The length of the absolute path name.
	996	1382	Path Name Indicator	Char(1)	The absolute path name indicator: **Y** The Absolute Path Name field contains an absolute path name for the object. **N** The Absolute Path Name field does not contain an absolute path name for the object.
	997	1383	Relative File ID[3]	Char(16)	The relative file ID of the absolute path name.
	1013	1399	Absolute Path Name[4]	Char(5002)	The absolute path name of the object.

Table 145. AD (Auditing Change) Journal Entries (continued). QASYADJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
[1]	These fields are used only for objects in the QOpenSys, "root" file systems, and user-defined file systems.				
[2]	An ID that has the left-most bit set and the rest of the bits zero indicates that the ID is NOT set.				
[3]	When the absolute path name indicator (offset 996) is ″N″, this field will contain the relative field ID of the path name. When the absolute path name indicator is ″Y″, this field will contain 16 bytes of hex zeroes.				
[4]	This is a variable length field. The first two bytes contain the length of the path name.				
[5]	If the object is in a library, this is the ASP information of the object's library. If the object is not in a library, this is the ASP information of the object.				

Table 146. AF (Authority Failure) Journal Entries. QASYAFJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.

Table 146. AF (Authority Failure) Journal Entries (continued). QASYAFJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
156	224	610	Violation Type[1]	Char(1)	**A** Not authorized to object
					B Restricted instruction
					C Validation failure (see offset 185)
					D Use of unsupported interface, object domain failure
					E Hardware storage protection error, program constant space violation
					F ICAPI authorization error
					G ICAPI authentication error
					I[7] System Java inheritance not allowed
					J Submit job profile error
					N Profile token not a regenerable token
					O Optical Object Authority Failure
					P Profile swap error
					R Hardware protection error
					S Default sign-on attempt
					T Not authorized to TCP/IP port
					U User permission request not valid
					V Profile token not valid for generating new profile token
					W Profile token not valid for swap
					X System violation — see offset 337 for violation codes
					Y Not authorized to the current JUID field during a clear JUID operation.
					Z Not authorized to the current JUID field during a set JUID operation.
157	225	611	Object Name [1, 5]	Char(10)	The name of the object.
167	235	621	Library Name	Char(10)	The name of the library the object is in.
177	245	631	Object Type	Char(8)	The type of object.

Table 146. AF (Authority Failure) Journal Entries (continued). QASYAFJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
185	253	639	Validation Error Action	Char(1)	Action taken after validation error detected, set only if the violation type (offset 156) is C. **A** The translation of the object was not attempted or it failed. The QALWOBJRST system value setting allowed the object to be restored. The user doing the restore did not have *ALLOBJ special authority and the system security level is set to 10, 20, or 30. Therefore, all authorities to the object were retained. **B** The translation of the object was not attempted or it failed. The QALWOBJRST system value setting allowed the object to be restored. The user doing the restore did not have *ALLOBJ special authority and the system security level is set to 40 or above. Therefore, all authorities to the object were revoked. **C** The translation of the object was successful. The translated copy was restored on the system. **D** The translation of the object was not attempted or it failed. The QALWOBJRST system value setting allowed the object to be restored. The user doing the restore had *ALLOBJ special authority. Therefore, all authorities to the object were retained. **E** System install time error detected. **F** The object was not restored because the signature is not OS/400 format.
186	254	640	Job Name	Char(10)	The name of the job.
196	264	650	User Name	Char(10)	The job user name.
206	274	660	Job Number	Zoned(6,0)	The job number.
212	280	666	Program Name	Char(10)	The name of the program.
222	290	676	Program Library	Char(10)	The name of the library where the program is found.
232	300	686	User Profile [2]	Char(10)	The name of the user that caused the authority failure.
242	310	696	Work Station Name	Char(10)	The name of the work station or work station type.
252	320	706	Program Instruction Number	Zoned(7,0)	The instruction number of the program.
259	327	713	Field name	Char(10)	The name of the field.

Table 146. AF (Authority Failure) Journal Entries (continued). QASYAFJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
269	337	723	Operation Violation Code	Char(3)	The type of operation violation that occurred, set only if the violation type (offset 224) is X. **HCA** Service tool user profile not authorized to perform hardware configuration operation (QYHCHCOP).
272	340	726	Office User	Char(10)	The name of the office user.
282	350	736	DLO Name	Char(12)	The name of the document library object.
294	362	748	(Reserved Area)	Char(8)	
302	370	756	Folder Path	Char(63)	The path of the folder.
365	433	819	Office on Behalf of User	Char(10)	User working on behalf of another user.
375			(Reserved Area)	Char(20)	
	443	829	(Reserved Area)	Char(18)	
	461	847	Object Name Length[3]	Binary(4)	The length of the object name.
395	463	849	Object Name CCSID[3]	Binary(5)	The coded character set identifier for the object name.
399	467	853	Object Name Country or Region ID[3]	Char(2)	The Country or Region ID for the object name.
401	469	855	Object Name Language ID[3]	Char(3)	The language ID for the object name.
404	472	858	(Reserved area)	Char(3)	
407	475	861	Parent File ID[3,4]	Char(16)	The file ID of the parent directory.
423	491	877	Object File ID[3,4]	Char(16)	The file ID of the object.
439	507	893	Object Name[3,6]	Char(512)	The name of the object.
	1019	1405	Object File ID	Char(16)	The file ID of the object.
	1035	1421	ASP Name[10]	Char(10)	The name of the ASP device.
	1045	1431	ASP Number[10]	Char(5)	The number of the ASP device.
	1050	1436	Path Name CCSID	Binary(5)	The coded character set identifier for the absolute path name.
	1054	1440	Path Name Country or Region ID	Char(2)	The Country or Region ID for the absolute path name.
	1056	1442	Path Name Language ID	Char(3)	The language ID for the the absolute path name.
	1059	1445	Path Name Length	Binary(4)	The length of the absolute path name.
	1061	1447	Path Name Indicator	Char(1)	The absolute path name indicator: **Y** The Absolute Path Name field contains an absolute path name for the object. **N** The Absolute Path Name field does not contain an absolute path name for the object.

Table 146. AF (Authority Failure) Journal Entries (continued). QASYAFJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	1062	1448	Relative File ID[8]	Char(16)	The relative file ID of the absolute path name.
	1078	1464	Absolute Path Name[9]	Char(5002)	The absolute path name of the object.
		6466	ASP program library name	Char(10)	ASP name for program library
		6476	ASP program library number	Char(5)	ASP number for program library

[1] When the violation type is for description ″G″, the object name contains the name of the *SRVPGM that contained the exit that detected the error. For more information about the violation types, see Table 117 on page 255.

[2] This field contains the name of the user that caused the entry. QSYS may be the user for the following:

- offsets 41 and 118 for *TYPE2 records
- offsets 55 and 132 for *TYPE4 records
- offsets 65 and 187 for *TYPE5 records

[3] These fields are used only for objects in the QOpenSys file system, the "root" file system, user-defined file systems, and QFileSvr.400.

[4] An ID that has the left-most bit set and the rest of the bits zero indicates that the ID is NOT set.

[5] When the violation type is ″T″, the object name contains the TCP/IP port the user is not authorized to use. The value is left justified and blank filled. The object library and object type fields will be blank.

[6] When the violation type is O, the optical object name is contained in the IFS object name field. The Country or Region ID, language ID, parent file ID, and object file ID fields will all contain blanks.

[7] The Java class object being created may not extend its base class because the base class has system Java attributes.

[8] When the absolute path name indicator (offset 1061) is ″N″, this field will contain the relative file ID of the path name. When the absolute path name indicator is ″Y″, this field will contain 16 bytes of hex zeroes.

[9] This is a variable length field. The first 2 bytes contain the length of the path name.

[10] If the object is in a library, this is the ASP information of the object's library. If the object is not in a library, this is the ASP information of the object.

Table 147. AP (Adopted Authority) Journal Entries. QASYAPJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	**S** Start **E** End **A** Adopted authority used during program activation
157	225	611	Object Name	Char(10)	The name of the program, service program, or SQL package
167	235	621	Library name	Char(10)	The name of the library.
177	245	631	Object Type	Char(8)	The type of object.

Table 147. AP (Adopted Authority) Journal Entries (continued). QASYAPJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
185	253	639	Owning User Profile	Char(10)	The name of the user profile whose authority is adopted.
195	263	649	Object File ID	Char(16)	The file ID of the object.
	279	665	ASP Name[1]	Char(10)	The name of the ASP device.
	289	675	ASP Number[1]	Char(5)	The number of the ASP device.

1 If the object is in a library, this is the ASP information of the object's library. If the object is not in a library, this is the ASP information of the object.

Table 148. AU (Attribute Changes) Journal Entries. QASYAUJ5 Field Description File

Offset			
J5	**Field**	**Format**	**Description**
610	Entry type	Char(1)	Type of entry
611	Action	Char(3)	Action
614	Name	Char(100)	Name
714	New value length	Binary(4)	New value length
716	New value CCSID	Binary(5)	New value CCSID
720	New value Country or Region ID	Char(2)	New value Country or Region ID
722	New value language ID	Char(3)	New value language ID
725	New value	Char(2002) [1]	New value
2727	Old value length	Binary(4)	Old value length
2729	Old value CCSID	Binary(5)	Old value CCSID
2733	Old value Country or Region ID	Char(2)	Old value Country or Region ID
2735	Old value language ID	Char(3)	Old value language ID
2738	Old value	Char(2002) [1]	Old value

1 This is a variable length field. The first 2 bytes contain the length of the field.

Table 149. CA (Authority Changes) Journal Entries. QASYCAJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **A** Changes to authority
157	225	611	Object Name	Char(10)	The name of the object.
167	235	621	Library Name	Char(10)	The name of the library the object is in.
177	245	631	Object Type	Char(8)	The type of object.
185	253	639	User Name	Char(10)	The name of the user profile whose authority is being granted or revoked.
195	263	649	Authorization List Name	Char(10)	The name of the authorization list.
					Authorities granted or removed:

Table 149. CA (Authority Changes) Journal Entries (continued). QASYCAJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
205	273	659	Object Existence	Char(1)	**Y** *OBJEXIST
206	274	660	Object Management	Char(1)	**Y** *OBJMGT
207	275	661	Object Operational	Char(1)	**Y** *OBJOPR
208	276	662	Authorization List Management	Char(1)	**Y** *AUTLMGT
209	277	663	Authorization List	Char(1)	**Y** *AUTL public authority
210	278	664	Read Authority	Char(1)	**Y** *READ
211	279	665	Add Authority	Char(1)	**Y** *ADD
212	280	666	Update Authority	Char(1)	**Y** *UPD
213	281	667	Delete Authority	Char(1)	**Y** *DLT
214	282	668	Exclude Authority	Char(1)	**Y** *EXCLUDE
215	283	669	Execute Authority	Char(1)	**Y** *EXECUTE
216	284	670	Object Alter Authority	Char(1)	**Y** *OBJALTER
217	285	671	Object Reference Authority	Char(1)	**Y** *OBJREF
218	286	672	(Reserved Area)	Char(4)	
222	290	676	Command Type	Char(3)	The type of command used. **GRT** Grant **RPL** Grant with replace **RVK** Revoke **USR** GRTUSRAUT operation
225	293	679	Field name	Char(10)	The name of the field.
235	303	689	(Reserved Area)	Char(10)	
245	313	699	Office User	Char(10)	The name of the office user.
255	323	709	DLO Name	Char(12)	The name of the DLO.
267	335	721	(Reserved Area)	Char(8)	
275	343	729	Folder Path	Char(63)	The path of the folder.
338	406	792	Office on Behalf of User	Char(10)	User working on behalf of another user.
348	416	802	Personal Status	Char(1)	**Y** Personal status changed
349	417	803	Access Code	Char(1)	**A** Access code added **R** Access code removed
350	418	804	Access Code	Char(4)	Access code.

Table 149. CA (Authority Changes) Journal Entries (continued). QASYCAJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
354			(Reserved Area)	Char(20)	
	422	808	(Reserved Area)	Char(18)	
	440	826	Object Name Length [1]	Binary(4)	The length of the object name.
374	442	828	Object Name CCSID[1]	Binary(5)	The coded character set identifier for the object name.
378	446	832	Object Name Country or Region ID[1]	Char(2)	The Country or Region ID for the object name.
380	448	834	Object Name Language ID[1]	Char(3)	The language ID for the object name.
383	451	837	(Reserved area)	Char(3)	
386	454	840	Parent File ID[1,2]	Char(16)	The file ID of the parent directory.
402	470	856	Object File ID[1,2]	Char(16)	The file ID of the object.
418	486	872	Object Name[1]	Char(512)	The name of the object.
	998	1384	Object File ID	Char(16)	The file ID of the object.
	1014	1400	ASP Name[5]	Char(10)	The name of the ASP device.
	1024	1410	ASP Number[5]	Char(5)	The number of the ASP device.
	1029	1415	Path Name CCSID	Binary(5)	The coded character set identifier for the absolute path name.
	1033	1419	Path Name Country or Region ID	Char(2)	The Country or Region ID for the absolute path name.
	1035	1421	Path Name Language ID	Char(3)	The language ID for the absolute path name.
	1038	1424	Path Name Length	Binary(4)	The length of the absolute path name.
	1040	1426	Path Name Indicator	Char(1)	The absolute path name indicator: **Y** The Absolute Path Name field contains an absolute path name for the object. **N** The Absolute Path Name field does not contain an absolute path name for the object.
	1041	1427	Relative File ID[3]	Char(16)	The relative file ID of the absolute path name.
	1057	1443	Absolute Path Name[4]	Char(5002)	The absolute path name of the object.

[1] These fields are used only for objects in the QOpenSys file system, the "root" file system, user-defined file systems, and QFileSvr.400.

[2] An ID that has the left-most bit set and the rest of the bits zero indicates that the ID is NOT set.

[3] When the path name indicator (offset 1040) is ″N″, this field will contain the relative file ID of the path name. When the path name indicator is ″Y″, this field will contain 16 bytes of hex zeroes.

[4] This is a variable length field. The first 2 bytes contain the length of the path name.

[5] If the object is in a library, this is the ASP information of the object's library. If the object is not in a library, this is the ASP information of the object.

Table 150. CD (Command String) Journal Entries. QASYCDJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry.
					C Command run
					L OCL statement
					O Operator control command
					P S/36 procedure
					S Command run after command substitution took place
					U Utility control statement
157	225	611	Object Name	Char(10)	The name of the object.
167	235	621	Library Name	Char(10)	The name of the library the object is in.
177	245	631	Object Type	Char(8)	The type of object.
185	253	639	Run from a CL program	Char(1)	**Y** Yes
					N No
186	254	640	Command string	Char(6000)	The command that was run, with parameters.
		6640	ASP name for command library	Char(10)	ASP name for command library
		6650	ASP number for command library	Char(5)	ASP number for command library

Table 151. CO (Create Object) Journal Entries. QASYCOJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry.
					N Create of new object
					R Replacement of existing object
157	225	611	Object Name	Char(10)	The name of the object.
167	235	621	Library Name	Char(10)	The name of the library the object is in.
177	245	631	Object Type	Char(8)	The type of object.
185	253	639	(Reserved Area)	Char(20)	
205	273	659	Office User	Char(10)	The name of the office user.
215	283	669	DLO Name	Char(12)	The name of the document library object created.
227	295	681	(Reserved Area)	Char(8)	
235	303	689	Folder Path	Char(63)	The path of the folder.

Table 151. CO (Create Object) Journal Entries (continued). QASYCOJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
298	366	752	Office on Behalf of User	Char(10)	User working on behalf of another user.
308			(Reserved Area)	Char(20)	
	376	762	(Reserved Area)	Char(18)	
	394	780	Object Name Length	Binary(4)	The length of the object name.
328	396	782	Object Name CCSID[1]	Binary(5)	The coded character set identifier for the object name.
332	400	786	Object Name Country or Region ID[1]	Char(2)	The Country or Region ID for the object name.
334	402	788	Object Name Language ID[1]	Char(3)	The language ID for the object name.
337	405	791	(Reserved area)	Char(3)	
340	408	794	Parent File ID[1,2]	Char(16)	The file ID of the parent directory.
356	424	810	Object File ID[1,2]	Char(16)	The file ID of the object.
372	440	826	Object Name[1]	Char(512)	The name of the object.
	952	1338	Object File ID	Char(16)	The file ID of the object.
	968	1354	ASP Name[5]	Char(10)	The name of the ASP device.
	978	1364	ASP Number[5]	Char(5)	The number of the ASP device.
	983	1369	Path Name CCSID	Binary(5)	The coded character set identifier for the absolute path name.
	987	1373	Path Name Country or Region ID	Char(2)	The Country or Region ID for the absolute path name.
	989	1375	Path Name Language ID	Char(3)	The language ID for the absolute path name.
	992	1378	Path Name Length	Binary(4)	The length of the absolute path name.
	994	1380	Path Name Indicator	Char(1)	The absolute path name indicator: **Y** The Absolute Path Name field contains an absolute path name for the object. **N** The Absolute Path Name field does not contain an absolute path name for the object.
	995	1381	Relative File ID[3]	Char(16)	The relative file ID of the absolute path name.
	1011	1397	Absolute Path Name[4]	Char(5002)	The absolute path name of the object.

Table 151. CO (Create Object) Journal Entries (continued). QASYCOJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**

[1] These fields are used only for objects in the QOpenSys, "root" file systems, and user-defined file systems.

[2] An ID that has the left-most bit set and the rest of the bits zero indicates that the ID is NOT set.

[3] When the path name indicator (offset 994) is ″N″, this field will contain the relative file ID of the path name. When the path name indicator is ″Y″, this field will contain 16 bytes of hex zeroes..

[4] This is a variable length field. The first 2 bytes contain the length of the path name.

[5] If the object is in a library, this is the ASP information of the object's library. If the object is not in a library, this is the ASP information of the object.

Table 152. CP (User Profile Changes) Journal Entries. QASYCPJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **A** Change to a user profile
157	225	611	User Profile Name	Char(10)	The name of the user profile that was changed.
167	235	621	Library Name	Char(10)	The name of the library.
177	245	631	Object Type	Char(8)	The type of object.
185	256	639	Command Name	Char(3)	The type of command used. **CRT** CRTUSRPRF **CHG** CHGUSRPRF **RST** RSTUSRPRF **DST** QSECOFR password reset using DST
188	256	642	Password Changed	Char(1)	**Y** Password changed
189	257	643	Password *NONE	Char(1)	**Y** Password is *NONE.
190	258	644	Password Expired	Char(1)	**Y** Password expired is *YES **N** Password expired is *NO
191	259	645	All Object Special Authority	Char(1)	**Y** *ALLOBJ special authority
192	260	646	Job Control Special Authority	Char(1)	**Y** *JOBCTL special authority
193	261	647	Save System Special Authority	Char(1)	**Y** *SAVSYS special authority
194	262	648	Security Administrator Special Authority	Char(1)	**Y** *SECADM special authority

Table 152. CP (User Profile Changes) Journal Entries (continued). QASYCPJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
195	263	649	Spool Control Special Authority	Char(1)	**Y** *SPLCTL special authority
196	264	650	Service Special Authority	Char(1)	**Y** *SERVICE special authority
197	265	651	Audit Special Authority	Char(1)	**Y** *AUDIT special authority
198	266	652	System Configuration Special Authority	Char(1)	**Y** *IOSYSCFG special authority
199	267	653	(Reserved Area)	Char(13)	
212	280	666	Group Profile	Char(10)	The name of a group profile.
222	290	676	Owner	Char(10)	Owner of objects created as a member of a group profile.
232	300	686	Group Authority	Char(10)	Group profile authority.
242	310	696	Initial Program	Char(10)	The name of the user's initial program.
252	320	706	Initial Program Library	Char(10)	The name of the library where the initial program is found.
262	330	716	Initial Menu	Char(10)	The name of the user's initial menu.
272	340	726	Initial Menu Library	Char(10)	The name of the library where the initial menu is found.
282	350	736	Current Library	Char(10)	The name of the user's current library.
292	360	746	Limited Capabilities	Char(10)	The value of limited capabilities parameter.
302	370	756	User Class	Char(10)	The user class of the user.
312	380	766	Priority Limit	Char(1)	The value of the priority limit parameter.
313	381	767	Profile Status	Char(10)	User profile status.
323	391	777	Group Authority Type	Char(10)	The value of the GRPAUTTYP parameter.
333	401	787	Supplemental Group Profiles	Char(150)	The names of up to 15 supplemental group profiles for the user.
483	551	937	User Identification	Char(10)	The uid for the user.
493	561	947	Group Identification	Char(10)	The gid for the user.

*Table 153. CQ (*CRQD Changes) Journal Entries.* QASYCQJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **A** Change to a *CRQD object
157	225	611	Object Name	Char(10)	The name of the object that was changed.
167	235	621	Library Name	Char(10)	The name of the object library.

*Table 153. CQ (*CRQD Changes) Journal Entries (continued).* QASYCQJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
177	245	631	Object Type	Char(8)	The type of object.
		639	ASP name	Char(10)	ASP name for CRQD library
		649	ASP number	Char(5)	ASP number for CRQD library

Table 154. CU (Cluster Operations) Journal Entries. QASYCUJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	1	1			Heading fields common to all entry types. See Table 141 on page 501 and Table 142 on page 503 for field listing.
	224	610	Entry Type	Char(1)	The type of entry.
					M Cluster control operation
					R Cluster Resource Group (*GRP) management operation
	225	611	Entry Action	Char(3)	The type of action.
					Add Add
					CRT Create
					DLT Delete
					DST Distribute
					END End
					FLO Fail over
					LST List information
					RMV Remove
					STR Start
					SWT Switch
					UPC Update attributes
	228	614	Status	Char(3)	The status of the request.
					ABN The request ended abnormally
					AUT Authority Failure, *IOSYSCFG is required
					END The request ended successfully
					STR The request was started
	231	617	CRG Object Name	Char(10)	The Cluster Resource Group object name.
					Note: This value is filled in when the entry type is R.
	241	627	CRG Library Name	Char(10)	The Cluster Resource Group object library.
					Note: This value is filled in when the entry type is R.
	251	637	Cluster Name	Char(10)	The name of the cluster.
	261	647	Node ID	Char(8)	The node ID.

Table 154. CU (Cluster Operations) Journal Entries (continued). QASYCUJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	269	655	Source Node ID	Char(8)	The source node ID.
	277	663	Source User Name	Char(10)	Name of the source system user that initiated the request.
	287	673	User Queue Name	Char(10)	Name of the user queue where responses are sent.
	297	683	User Queue Library	Char(10)	The user queue library.
		693	ASP name	Char(10)	ASP name for user queue library
		703	ASP number	Char(5)	ASP number for user queue library

Table 155. CV (Connection Verification) Journal Entries. QASYCVJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	1	1			Heading fields common to all entry types. See Table 141 on page 501 and Table 142 on page 503 for field listing.
	224	610	Entry Type	Char(1)	The type of entry.
					C Connection established
					E Connection ended
					R Connection rejected
	225	611	Action	Char(1)	Action taken for the connection type.
					" " Connection established or ended normally. Used for Entry Type C or E.
					A Peer was not authenticated. Used for Entry Type E or R.
					C No response from the authentication server. Used for Entry Type R.
					L LCP configuration error. Used for Entry Type R.
					N NCP configuration error. Used for Entry Type R.
					P Password is not valid. Used for Entry Type E or R.
					R Authentication was rejected by peer. Used for Entry Type R.
					T L2TP configuration error. Used for Entry Type E or R.
					U User is not valid. Used for Entry Type E or R.
	226	612	Point to Point Profile Name	Char(10)	The point to point profile name.

Table 155. CV (Connection Verification) Journal Entries (continued). QASYCVJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	236	622	Protocol	Char(10)	The type of entry. **L2TP** Layer 2 Tunneling protocol **PPP** Point to Point protocol. **SLIP** Serial Line Internet Protocol.
	246	632	Local Authentication Method	Char(10)	The type of entry. **CHAP** Challenge Handshake Authentication Protocol. **PAP** Password Authentication Protocol. **SCRIPT** Script method.
	256	642	Remote Authentication Method	Char(10)	The type of entry. **CHAP** Challenge Handshake Authentication Protocol. **PAP** Password Authentication Protocol. **RADIUS** Radius method. **SCRIPT** Script method.
	266	652	Object Name	Char(10)	The *VLDL object name.
	276	662	Library Name	Char(10)	The *VLDL object library name.
	286	672	*VLDL User Name	Char(100)	The *VLDL user name.
	386	772	Local IP Address	Char(40)	The local IP address.
	426	812	Remote IP Address	Char(40)	The remote IP address.
	466	852	IP forwarding	Char(1)	The type of entry. **Y** IP forwarding is on. **N** IP forwarding is off.

Table 155. CV (Connection Verification) Journal Entries (continued). QASYCVJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	467	853	Proxy ARP	Char(1)	The type of entry.
					Y Proxy ARP is enabled.
					N Proxy ARP is not enabled.
	468	854	Radius Name	Char(10)	The AAA profile name.
	478	864	Authenticating IP Address	Char(40)	The authenticating IP address.
	518	904	Account Sesion ID	Char(14)	The account session ID.
	532	918	Account Multi-Session ID	Char(14)	The account multi-session ID.
	546	932	Account Link Count	Binary(4)	The account link count.
	548	934	Tunnel Type	Char(1)	The tunnel type:
					0 Not tunneled
					3 L2TP
					6 AH
					9 ESP
	549	935	Tunnel Client Endpoint	Char(40)	Tunnel client endpoint.
	589	975	Tunnel Server Endpoint	Char(40)	Tunnel server endpoint.
	629	1015	Account Session Time	Char(8)	The account session time. Used for Entry Type E or R.
	637	1023	Account Terminate Cause	Binary(4)	The account terminate cause. Used for Entry Type E or R.
		1025	ASP name	Char(10)	ASP name for validation list library
		1035	ASP number	Char(5)	ASP number for validation list library

Table 156. CY (Cryptographic Configuration) Journal Entries. QASYCYJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
	224	610	Entry Type	Char(1)	The type of entry.
					A Access Control function
					F Facility Control function
					M Master Key function

Table 156. CY (Cryptographic Configuration) Journal Entries (continued). QASYCYJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	225	611	Action	Char(3)	The cryptographic configuration function performed: **CCP** Define a card profile. **CCR** Define a card role. **CLK** Set clock. **CLR** Clear master keys. **CRT** Create master keys. **DCP** Delete a card profile. **DCR** Delete a card role. **DST** Distribute master keys. **EID** Set environment ID. **FCV** Load/clear FCV. **INI** Reinitialize card.. **QRY** Query role or profile information. **RCP** Replace a card profile. **RCR** Replace a card role. **RCV** Receive master keys. **SET** Set master keys. **SHR** Cloning shares.
	228	614	Card profile	Char(8)	The name of the card profile..
	236	622	Card Role	Char(8)	The role of the card profile.
	244	630	Device Name	Char(10)	The name of the cryptographic device.

Table 157. DI (Directory Services) Journal Entries. QASYDIJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
	224	610	Entry Type	Char(1)	The type of entry. **L** LDAP Operation

Table 157. DI (Directory Services) Journal Entries (continued). QASYDIJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	225	611	Operation Type	Char(2)	The type of LDAP operation:
					AD Audit attribute change.
					AF Authority failure.
					BN Successful bind.
					CA Object authority change.
					CF Configuration change.
					CO Object create.
					CP Password change.
					DO Object delete.
					EX LDAP directory export.
					IM LDAP directory import.
					OM Object management (rename).
					OW Ownership change.
					PW Password fail.
					UB Successful unbind.
					ZC Object change.
					ZR Object read.
	227	613	Authority Failure Code	Char(1)	Code for authority failures. This field is used only if the operation type (offset 225) is AF.
					A Unauthorized attempt to change audit value.
					B Unauthorized bind attempt.
					C Unauthorized object create attempt.
					D Unauthorized object delete attempt.
					E Unauthorized export attempt.
					F Unauthorized configuration change (administrator, change log, backend library, replicas replicas, publishing)
					I Unauthorized import attempt.
					M Unauthorized modify attempt.
					R Unauthorized read (search) attempt.
	228	614	Configuration Change	Char(1)	Configuration changes. This field is only used if the operation type (offset 225) is CF.
					A Administrator ND change
					C Change log on/off
					L Backend library name change
					P Publishing agent change
					R Replica server change

Table 157. DI (Directory Services) Journal Entries (continued). QASYDIJ4/J5 Field Description File

Offset					
JE	J4	J5	Field	Format	Description
	229	615	Configuration Change Code	Char(1)	Code for configuration changes. This field is used only if the operation type (offset 225) is CF. **A** Item added to configuration **D** Item deleted from configuration **M** Item modified
	230	616	Propagate Flag	Char(1)	Indicates the new setting of the owner or ACL propagate value. This field is used only if the operation type (offset 225) is CA or OW. **T** True **F** False
	231	617	Bind Authentication Choice	Char(20)	The bind authentication choice. This field is used only if the operation type (offset 225) is BN.
	251	637	LDAP Version	Char(4)	Version of client making request. This field is used only if the operation was done through the LDAP server. **2** LDAP Version 2 **3** LDAP Version 3
	255	641	SSL Indicator	Char(1)	Indicates if SSL was used on the request. This field is used ony if the operation was done through the LDAP server. **0** No **1** Yes
	256	642	Request Type	Char(1)	The type of request. This field is used only if the operation was done through the LDAP server. **A** Authenticated **N** Anonymous **U** Unauthenticated
	257	643	Connection ID	Char(20)	Connection ID of the request. This field is used only if the operation was done through the LDAP server.
	277	663	Client IP Address	Char(50)	IP address and port number of the client request. This field is used only if the operation was done through the LDAP server.
	327	713	User Name CCSID	Bin(5)	The coded character set identifier of the user name.
	331	717	User Name Length	Bin(4)	The length of the user name.
	333	719	User Name[1]	Char(2002)	The name of the LDAP user.
	2335	2721	Object Name CCSID	Bin(5)	The coded character set identifier of the object name.
	2339	2725	Object Name Length	Bin(4)	The length of the object name.
	2341	2727	Object Name[1]	Char(2002)	The name of the LDAP object.
	4343	4729	Owner Name CCSID	Bin(5)	The coded character set identifier of the owner name. This field is used only if the operation type (offset 225) is OW.

Table 157. DI (Directory Services) Journal Entries (continued). QASYDIJ4/J5 Field Description File

Offset					
JE	J4	J5	Field	Format	Description
	4347	4733	Owner Name Length	Bin(4)	The length of the owner name. This field is used only if the operation type is OW.
	4349	4735	Owner Name[1]	Char(2002)	The name of the owner. This field is used only if the operation type (offset 225) is OW.
	6351	6737	New Name CCSID	Bin(5)	The coded character set identifier of the new name. This field is used only if the operation type (offset 225) is OM, OW, ZC, or AF+M. • For operation type OM, this field will contain the CCSID of the new object name. • For operation type OW, this field will contain the CCSID of the new owner name. • For operation types ZC or AF+M, this field will contain the CCSID of the list of changed attribute types in the New Name field.
	6355	6741	New Name Length	Bin(4)	The length of the new name. This field is used only if the operation type (offset 225) is OM, OW, ZC, or AF+M. • For operation type OM, this field will contain the length of the new object name. • For operation type OW, this field will contain the length of the new owner name. • For operation types ZC or AF+M, this field will contain the length of the list of changed attribute types in the New Name field.
	6357	6743	New Name[1]	Char(2002)	The new name. This field is used only if the operation type (offset 225) is OM, OW, ZC, or AF+M. • For operation type OM, this field will contain the new object name. • For operation type OW, this field will contain the new owner name. • For operation types ZC or AF+M, this field will contain a list of changed attribute types.
	8359	8745	Object File ID[2]	Char(16)	The file ID of the object for export.
	8375	8761	ASP Name[2]	Char(10)	The name of the ASP device.
	8385	8771	ASP Number[2]	Char(5)	The number of the ASP device.
	8390	8776	Path Name CCSID[2]	Bin(5)	The coded character set identifier of the absolute path name.
	8394	8780	Path Name Country or Region ID[2]	Char(2)	The Country or Region ID of the absolute path name.
	8396	8782	Path Name Language ID[2]	Char(3)	The language ID of the absolute path name.
	8399	8785	Path Name Length[2]	Bin(4)	The length of the absolute path name.
	8401	8787	Path Name Indicator[2]	Char(1)	The absolute path name indicator. **Y** The Absolute Path Name field contains an absolute path name for the object. **N** The Absolute Path Name field does not contain an absolute path name for the object.

Table 157. DI (Directory Services) Journal Entries (continued). QASYDIJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	8402	8788	Relative File ID[2,3]	Char(16)	The relative file ID of the absolute path name.
	8418	8804	Absolute Path Name[1,2]	Char(5002)	The absolute path name of the object.

[1] This is a variable length field. The first 2 bytes contain the length of the value in the field.

[2] These fields are used only if the operation type (offset 225) is EX or IM.

[3] When the path name indicator (offset 8401) is ″N″, this field will contain the relative file ID of the path name. When the path name indicator is ″Y″, this field will contain 16 bytes of hex zeroes.

Table 158. DO (Delete Operation) Journal Entries. QASYDOJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry.
					A Object was deleted not under commitment control)
					C A pending object delete was committed
					D A pending object create was rolled back
					P The object delete is pending (the delete was performed under commitment control)
					R A pending object delete was rolled back
157	225	611	Object Name	Char(10)	The name of the object.
167	235	621	Library Name	Char(10)	The name of the library the object is in.
177	245	631	Object Type	Char(8)	The type of object.
185	253	639	(Reserved Area)	Char(20)	
205	273	659	Office User	Char(10)	The name of the office user.
215	283	669	DLO Name	Char(12)	The name of the document library object.
227	295	681	(Reserved Area)	Char(8)	
235	303	689	Folder Path	Char(63)	The path of the folder.
298	366	752	Office on Behalf of User	Char(10)	User working on behalf of another user.
308			(Reserved Area)	Char(20)	
	376	762	(Reserved Area)	Char(18)	
	394	780	Object Name Length [1]	Binary(4)	The length of the object name.
328	396	782	Object Name CCSID[1]	Binary(5)	The coded character set identifier for the object name.

Table 158. DO (Delete Operation) Journal Entries (continued). QASYDOJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
332	400	786	Object Name Country or Region ID[1]	Char(2)	The Country or Region ID for the object name.
334	402	788	Object Name Language ID[1]	Char(3)	The language ID for the object name.
337	405	791	(Reserved area)	Char(3)	
340	408	794	Parent File ID[1,2]	Char(16)	The file ID of the parent directory.
356	424	810	Object File ID[1,2]	Char(16)	The file ID of the object.
372	440	826	Object Name[1]	Char(512)	The name of the object.
	952	1338	Object File ID	Char(16)	The file ID of the object.
	968	1354	ASP Name[5]	Char(10)	The name of the ASP device.
	978	1364	ASP Number[5]	Char(5)	The number of the ASP device.
	983	1369	Path Name CCSID	Binary5)	The coded character set identifier for the absolute path name.
	987	1373	Path Name Country or Region ID	Char(2)	The Country or Region ID for the absolute path name.
	989	1375	Path Name Language ID	Char(3)	The language ID for the absolute path name.
	992	1378	Path Name Length	Binary(4)	The length of the absolute path name.
	994	1380	Path Name Indicator	Char(1)	The absolute path name indicator: **Y** The Absolute Path Name field contains an absolute path name for the object. **N** The Absolute Path Name field does not contain an absolute path name for the object.
	995	1381	Relative File ID[3]	Char(16)	The relative file ID of the absolute path name.
	1011	1397	Absolute Path Name[4]	Char(5002)	The absolute path name of the object.

[1] These fields are used only for objects in the QOpenSys, "root" file systems, and user-defined file systems.

[2] An ID that has the left-most bit set and the rest of the bits zero indicates that the ID is NOT set.

[3] When the path name indicator (offset 994) is ″N″, this field will contain the relative file ID of the path name. When the path name indicator is ″Y″, this field will contain 16 bytes of hex zeroes.

[4] This is a variable length field. The first 2 bytes contain the length of the path name.

[5] If the object is in a library, this is the ASP information of the object's library. If the object is not in a library, this is the ASP information of the object.

Table 159. DS (IBM-Supplied Service Tools User ID Reset) Journal Entries. QASYDSJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **A** Reset of a service tools user ID password. **C** Changed to a service tools user ID. **P** Service tools user ID password was changed.
157	225	611	IBM-Supplied Service Tools User ID Reset	Char(1)	**Y** Request to reset an IBM-supplied service tools user ID.
158	226	612	Service Tools User ID Type	Char(10)	The type of service tools user ID ***SECURITY** ***FULL** ***BASIC**
168	236	622	Service Tools User ID New Name	Char(8)	The name of the service tools user ID.
176	244	630	Service Tools User ID Password Change	Char(1)	Request to change the service tools user ID password. **Y** Request to change service tools user ID password.
	245	631	Service Tools User ID New Name	Char(10)	The name of the service tools user ID.
	255	641	Service Tools User ID Requesting Profile	Char(10)	The name of the service tools user ID that requested the change.

Table 160. EV (Environment Variable) Journal Entries. QASYEVJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
	224	610	Entry Type	Char(1)	The type of entry. **A** Add **C** Change **D** Delete

Table 160. EV (Environment Variable) Journal Entries (continued). QASYEVJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	225	611	Name Truncated	Char(1)	Indicates whether the environment variable name (offset 232), is truncated.
					Y Environment variable name truncated.
					N Environment variable name not truncated.
	226	612	CCSID	Binary(5)	The CCSID of the environment variable name.
	230	616	Length	Binary(4)	The length of the environment variable name.
	232	618	Environment Variable Name[2]	Char(1002)	The name of the environment variable.
	1234	1620	New Name Truncated[1]	Char(1)	Indicates whether the new environment variable name (offset 1241), is truncated.
					Y Environment variable value truncated.
					N Environment variable value not truncated.
	1235	1621	New Name CCSID[1]	Binary(5)	The CCSID of the new environment variable name.
	1239	1625	New Name Length[1]	Binary(4)	The length of the new environment variable name.
	1241	1627	New Environment Variable Name[1,2]	Char (1002)	The new environment variable name.

[1] These fields are used when the entry type is C.

[2] This is a variable length field. The first two bytes contain the length of the environment variable name.

Table 161. GR (Generic Record) Journal Entries. QASYGRJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	1	1			Heading fields common to all entry types. See Table 141 on page 501 and Table 142 on page 503 for field listing.
	224	610	Entry Type	Char(1)	The type of entry.
					A Exit program added
					D Exit program removed
					F Function registration operations
					R Exit program replaced
	225	611	Action	Char(2)	The action performed.
					ZC Change
					ZR Read
	227	613	User Name	Char(10)	User profile name For entry type F, this field contains the name of the user the function registration operation was performed against.
	237	623	Field 1 CCSID	Binary (5)	The CCSID value for field 1.

Table 161. GR (Generic Record) Journal Entries (continued). QASYGRJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	241	627	Field 1 Length	Binary (4)	The length of the data in field 1.
	243	629	Field 1	Char(102)[1]	Field 1 data For entry type F, this field contains the description of the function registration operation that was performed. The possible values are: ***REGISTER:** Function has been registered ***REREGISTER:** Function has been updated ***DEREGISTER:** Function has been de-registered ***CHGUSAGE:** Function usage information has changed ***CHKUSAGE:** Function usage was checked for a user and the check passed ***USAGEFAILURE:** Function usage was checked for a user and the check failed For entry types A, D, and R, this field will contain the exit program information for the specific function that was performed.
	345	731	Field 2 CCSID	Binary (5)	The CCSID value for field 2.
	349	735	Field 2 Length	Binary (4)	The length of the data in field 2.
	351	737	Field 2	Char (102)[1]	Field 2 data For entry type F, this field contains the name of the function that was operated on.
	453	839	Field 3 CCSID	Binary (5)	The CCSID value for field 3.
	457	843	Field 3 Length	Binary (4)	The length of the data in field 3.

Table 161. GR (Generic Record) Journal Entries (continued). QASYGRJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	459	845	Field 3	Char(102)[1]	Field 3 data. For entry type F, this field contains the usage setting for a user. There is a value for this field only if the function registration operation is one of the following: ***REGISTER:** When the operation is *REGISTER, this field contains the default usage value. The user name will be *DEFAULT. ***REREGISTER:** When the operation is *REREGISTER, this field contains the default usage value. The user name will be *DEFAULT. ***CHGUSAGE:** When the operation is *CHGUSAGE, this field contains the usage value for the user specified in the user name field.
	561	947	Field 4 CCSID	Binary (5)	The CCSID value for field 4.
	565	951	Field 4 Length	Binary (4)	The length of the data in field 4.
	567	953	Field 4	Char(102)[1]	Field 4 data. For entry type F, this field contains the allow *ALLOBJ setting for the function. There is a value for this field only if the function registration operation is one of the following: ***REGISTER** ***REREGISTER**

[1] This is a variable length field. The first 2 bytes contain the length of the field.

Table 162. GS (Give Descriptor) Journal Entries. QASYGSJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **G** Give descriptor **R** Received descriptor **U** Unable to use descriptor
157	225	611	Job Name	Char(10)	The name of the job.
167	235	621	User Name	Char(10)	The name of the user.
177	245	631	Job Number	Zoned (6,0)	The number of the job.

Table 162. GS (Give Descriptor) Journal Entries (continued). QASYGSJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
183	251	637	User Profile Name	Char (10)	The name of the user profile.
	261	647	JUID	Char (10)	The Job User IDentity of the target job. (This value applies only to subtype G audit records.)

Table 163. IP (Interprocess Communication) Journal Entries. QASYIPJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry.
					A Ownership and/or authority changes
					C create
					D Delete
					F Authority failure
					G Get
					M Shared memory attach
					Z Normal semaphore close or shared memory detach
157	225	611	IPC Type	Char(1)	IPC Type
					M Shared memory
					N Normal semaphore
					Q Message queue
					S Semaphore
158	226	612	IPC Handle	Binary(5)	IPC handle ID
162	230	616	New Owner	Char(10)	New owner of IPC entity
172	240	626	Old Owner	Char(10)	Old owner of IPC entity
182	250	636	Owner Authority	Char(3)	Owner's authority to IPC entity
					***R** read
					***W** write
					***RW** read and write
185	253	639	New Group	Char(10)	Group associated with IPC entity
195	263	649	Old Group	Char(10)	Previous group associated with IPC entity
205	273	659	Group Authority	Char(3)	Group's authority to IPC entity
					***R** read
					***W** write
					***RW** read and write

Table 163. IP (Interprocess Communication) Journal Entries (continued). QASYIPJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
208	276	662	Public Authority	Char(3)	Public's authority to IPC entity ***R** read ***W** write ***RW** read and write
211	279	665	CCSID Semaphore Name	Binary(5)	The CCSID of the semaphore name.
216	283	669	Length Semaphore Name	Binary(4)	The length of the semaphore name.
218	285	671	Semaphore Name	Char(2050)	The semaphore name. **Note:** This is a variable length field. The first 2 characters contain the length of the semaphore name.

Table 164. IR (IP Rules Actions) Journal Entries. QASYIRJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	1	1			Heading fields common to all entry types. See Table 141 on page 501 andTable 142 on page 503 for field listing.
	224	610	Entry Type	Char(1)	The type of entry. **L** IP rules have been loaded from a file. **N** IP rules have been unloaded for an IP Security connection **P** IP rules have been loaded for an IP Security connection **R** IP rules have been read and copied to a file. **U** IP rules have been unloaded (removed).
	225	611	File Name	Char(10)	The name of the QSYS file used to load or receive the IP rules. This value is blank if the file used was not in the QSYS file system.
	235	621	File Library	Char(10)	The name of the QSYS file library.
	245	631	Reserved	Char(18)	
	263	649	File Name Length	Binary (4)	The length of the file name.
	265	651	File Name CCSID[1]	Binary (5)	The coded character set identifier for the file name.
	269	655	File Country or Region ID[1]	Char(2)	The Country or Region ID for the file name.
	271	657	File Language ID[1]	Char(3)	The language ID for the file name.
	274	660	Reserved	Char(3)	

Table 164. IR (IP Rules Actions) Journal Entries (continued). QASYIRJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	277	663	Parent File ID[1, 2]	Char(16)	The file ID of the parent directory.
	293	679	Object File ID[1, 2]	Char(16)	The file ID of the file.
	309	695	File Name[1]	Char(512)	The name of the file.
	821	1207	Connection sequence	Char(40)	The connection name.
	861	1247	Object File ID	Char(16)	The file ID of the object.
	877	1263	ASP Name	Char(10)	The name of the ASP device.
	887	1273	ASP Number[5]	Char(5)	The number of the ASP device.
	892	1278	Path Name CCSID	Binary(5)	The coded character set identifier for the absolute path name.
	896	1282	Path Name Country or Region ID	Char(2)	The Country or Region ID for the absolute path name.
	898	1284	Path Name Language ID	Char(3)	The language ID for the absolute path name.
	901	1287	Path Name Length	Binary(4)	The length of the absolute path name.
	903	1289	Path Name Indicator	Char(1)	The absolute path name indicator: **Y** The Absolute Path Name field contains an absolute path name for the object. **N** The Absolute Path Name field does not contain an absolute path name for the object.
	904	1290	Relative File ID[3]	Char(16)	The relative file ID of the absolute path name.
	920	1306	Absolute Path Name[4]	Char(5002)	The absolute path name of the object.

1 These fields are used only for objects in the QOpenSys file system and the 'root' file system.

2 If the ID has the left-most bit set and the rest of the bits zero, the ID is **not** set.

3 When the path name indicator (offset 903) is "N" this field will contain the relative file ID of the path name. When the path name indicator is "Y", this field will contain 16 bytes of hex zeroes..

4 This is a variable length field. The first two bytes contain the length of the field.

5 If the object is in a library, this is the ASP information of the object's library. If the object is not in a library, this is the ASP information of the object.

Table 165. IS (Internet Security Management) Journal Entries. QASYISJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	1	1			Heading fields common to all entry types. See Table 141 on page 501 and Table 142 on page 503 for field listing.

Table 165. IS (Internet Security Management) Journal Entries (continued). QASYISJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	224	610	Entry Type	Char(1)	The type of entry. **A** Fail (this type no longer used) **C** Normal (this type no longer used) **U** Mobile User (this type no longer used) **1** IKE Phase 1 SA Negotiation **2** IKE Phase 2 SA Negotiation
	225	611	Local IP Address	Char(15)	Local IP Address.
	240	626	Local Client ID Port	Char(5)	Local Client ID port.
	245	631	Remote IP Address	Char (15)	Remote IP address.
	260	646	Remote Client ID Port	Char (5)	Remote Client ID Port (valid for phase 2).
	265	651	Mobile ID	Char (256)	Mobile ID. This field no longer used.
	521	907	Result Code	Char(4)	Negotiation Result: **0** Successful **1–30** Protocol specific errors (documented in ISAKMP RFC2408, found at: http://www.ietf.org) **82xx** iSeries VPN Key Manager specific errors
	525	911	CCSID	Bin(5)	The coded character set identifier for the following fields: • Local ID • Local Client ID Value • Remote ID • Remote Client ID Value
	529	915	Local ID	Char(256)	Local IKE identifier
	785	1171	Local Client ID Type	Char(2)	Type of client ID (valid for phase 2): **1** IP version 4 address **2** Fully qualified domain name **3** User fully qualified domain name **4** IP version 4 subnet **7** IP version 4 address range **9** Distinguished name **11** Key identifier
	787	1173	Local Client ID Value	Char(256)	Local client ID (valid for phase 2)
	1043	1429	Local Client ID Protocol	Char(4)	Local client ID protocol (valid for phase 2)
	1047	1433	Remote ID	Char(256)	Remote IKE identifier

Table 165. IS (Internet Security Management) Journal Entries (continued). QASYISJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	1303	1689	Remote Client ID Type	Char(2)	Type of client ID (valid for phase 2) **1** IP version 4 address **2** Fully qualified domain name **3** User fully qualified domain name **4** IP version 4 subnet **7** IP version 4 address range **9** Distinguished name **11** Key identifier
	1305	1691	Remote Client ID Value	Char(256)	Remote client ID (valid for phase 2)
	1561	1947	Remote Client ID Protocol	Char(4)	Remote client ID protocol (valid for phase 2)

Table 166. JD (Job Description Change) Journal Entries. QASYJDJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **A** User profile specified for the USER parameter of a job description
157	225	611	Job Description	Char(10)	The name of the job description that had the USER parameter changed.
167	235	621	Library Name	Char(10)	The name of the library the object is in.
177	245	631	Object Type	Char(8)	The type of object.
185	253	639	Command Type	Char(3)	The type of command used. **CHG** Change Job Description (CHGJOBD) command. **CRT** Create Job Description (CRTJOBD) command.
188	256	642	Old User	Char(10)	The name of the user profile specified for the USER parameter before the job description was changed.
198	266	652	New User	Char(10)	The name of the user profile specified for the user parameter when the job description was changed.
		662	ASP name	Char(10)	ASP name for JOBD library
		672	ASP number	Char(5)	ASP number for JOBD library

Table 167. JS (Job Change) Journal Entries. QASYJSJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	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 Attach prestart or batch immediate job
					Q Change query attributes
					R Release
					S Start
					T Modify profile or group profile using a profile token.
					U CHGUSRTRC
					V Virtual device changed by QWSACCDS API.
157	225	611	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

Table 167. JS (Job Change) Journal Entries (continued). QASYJSJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
158	226	612	Job Subtype	Char(1)	The subtype of the job.
					' ' No subtype
					D Batch immediate
					E Procedure start request
					J Prestart
					P Print driver
					Q Query
					T MRT
					U Alternate spool user
159	227	613	Job Name	Char(10)	The first part of the qualified job name being operated on
169	237	623	Job User Name	Char(10)	The second part of the qualified job name being operated on
179	247	633	Job Number	Char(6)	The third part of the qualified job name being operated on
185	253	639	Device Name	Char(10)	The name of the device
195	263	649	Effective User Profile[2]	Char(10)	The name of the effective user profile for the thread
205	273	659	Job Description Name	Char(10)	The name of the job description for the job
215	283	669	Job Description Library	Char(10)	The name of the library for the job description
225	293	679	Job Queue Name	Char(10)	The name of the job queue for the job
235	303	689	Job Queue Library	Char(10)	The name of the library for the job queue
245	313	699	Output Queue Name	Char(10)	The name of the output queue for the job
255	323	709	Output Queue Library	Char(10)	The name of the library for the output queue
265	333	719	Printer Device	Char(10)	The name of the printer device for the job
275	343	729	Library List[2]	Char(430)	The library list for the job
705	773	1159	Effective Group Profile Name[2]	Char(10)	The name of the effective group profile for the thread
715	783	1169	Supplemental Group Profiles[2]	Char(150)	The names of the supplemental group profiles for the thread.
	933	1319	JUID Description	Char(1)	Describes the meaning of the JUID field:
					' ' The JUID field contains the value for the JOB.
					C The clear JUID API was called. The JUID field contains the new value.
					S The set JUID API was called. The JUID field contains the new value.
	934	1320	JUID Field	Char(10)	Contains the JUID value
	944	1330	Real User Profile	Char(10)	The name of the real user profile for the thread.
	954	1340	Saved User Profile	Char(10)	The name of the saved user profile for the thread.

Table 167. JS (Job Change) Journal Entries (continued). QASYJSJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	964	1350	Real Group Profile	Char(10)	The name of the real group profile profile for the thread.
	974	1360	Saved Group Profile	Char(10)	The name of the saved group profile profile for the thread.
	984	1370	Real User Changed[3]	Char(1)	The real user profile was changed. **Y** Yes **N** No
	985	1371	Effective User Changed[3]	Char(1)	The effective user profile was changed. **Y** Yes **N** No
	986	1372	Saved User Changed[3]	Char(1)	The saved user profile was changed **Y** Yes **N** No
	987	1373	Real Group Changed[3]	Char(1)	The real group profile was changed. **Y** Yes **N** No
	988	1374	Effective Group Changed[3]	Char(1)	The effective group profile was changed **Y** Yes **N** No
	989	1375	Saved Group Changed[3]	Char(1)	The saved group profile was changed. **Y** Yes **N** No
	990	1376	Supplemental Groups Changed[3]	Char(1)	The supplemental group profiles were changed. **Y** Yes **N** No
	991	1377	Library list Number[4]	Bin(4)	The number of libraries in the library list extension field (offset 993).
	993	1379	Library List Extension[4,5]	Char(2252)	The extension to the library list for the job.
		3631	Library ASP group	Char(10)	Library ASP group
		3641	ASP name	Char(10)	ASP name for JOBD library
		3651	ASP number	Char(5)	ASP number for JOBD library

[1] This field is blank if the job is on the job queue and has not run.

[2] When the JS audit record is generated because one job performs an operation on another job then this field will contain data from the initial thread of the job that is being operated on. In all other cases, the field will contain data from the thread that performed the operation.

[3] This field is used only when entry type (offset 224) is M or T.

[4] This field is used only if the number of libraries in the library list exceeds the size of the field at offset 343.

[5] This is a variable length field. The first two bytes contain the length of the data in the field.

Table 168. KF (Key Ring File) Journal Entries. QASYKFJ4/J5 Field Description File

Offset					
JE	J4	J5	Field	Format	Description
	1	1			Heading fields common to all entry types. See Table 141 on page 501 andTable 142 on page 503 for field listing.
	224	610	Entry Type	Char(1)	The type of entry.
					C Certificate operation
					K Key ring file operation
					P Password incorrect
					T Trusted root operation
	225	611	Certificate Operation	Char(3)	Type of action[4].
					ADK Certificate with private key added
					ADD Certificate added
					REQ Certificate requested
					SGN Certificate signed
	228	614	Key Ring Operation	Char(3)	Type of action[5].
					ADD Key ring pair added
					DFT Key ring pair designated as default.
					EXP Key ring pair exported
					IMP Key ring pair imported
					LST List the key ring pair labels in a file
					PWD Change key ring file password
					RMV Key ring pair removed
					INF Key ring pair information retrieval
					2DB Key ring file converted to key database file format
					2YR Key database file converted to key ring file
	231	617	Trusted Root Operation	Char(3)	Type of action[6].
					TRS Key ring pair designated as trusted root
					RMV Trusted root designation removed
					LST List trusted roots
	234	620	Reserved	Char(18)	
	252	638	Object Name Length	Binary(4)	Key ring file name length.
	254	640	Object Name CCSID	Binary(5)	Key ring file name CCSID.
	258	644	Object Name Country or Region ID	Char(2)	Key ring file name Country or Region ID.
	260	646	Object Name Language ID	Char(3)	Key ring file name language ID.
	263	649	Reserved	Char(3)	
	266	652	Parent File ID	Char(16)	Key ring parent directory file ID.

Table 168. KF (Key Ring File) Journal Entries (continued). QASYKFJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	282	668	Object File ID	Char(16)	Key ring directory file name.
	298	684	Object Name	Char(512)	Key ring file name.
	810	1196	Reserved	Char(18)	
	828	1214	Object Name length	Binary(4)	Source or target file name length.
	830	1216	Object Name CCSID	Binary(5)	Source or target file name CCSID.
	834	1220	Object Name Country or Region ID	Char(2)	Source or target file name Country or Region ID.
	836	1222	Object Name Language ID	Char(3)	Source or target file name language ID.
	839	1225	Reserved	Char(3)	
	842	1228	Parent File ID	Char(16)	Source or target parent directory file ID.
	858	1244	Object File ID	Char(16)	Source or target directory file ID.
	874	1260	Object Name	Char(512)	Source or target file name.
	1386	1772	Certificate Label Length	Binary(4)	The length of the certificate label.
	1388	1774	Certificate Label[1]	Char(1026)	The certificate label.
	2414	2800	Object File ID	Char(16)	The file ID of the key ring file.
	2430	2816	ASP Name	Char(10)	The name of the ASP device.
	2440	2826	ASP Number	Char(5)	The number of the ASP device.
	2445	2831	Path Name CCSID	Binary(5)	The coded character set identifier for the absolute path name.
	2449	2835	Path Name Country or Region ID	Char(2)	The Country or Region ID for the absolute path name.
	2451	2837	Path Name Language ID	Char(3)	The language ID for the absolute path name.
	2454	2840	Path Name Length	Binary(4)	The length of the absolute path name.
	2456	2842	Path Name Indicator	Char(1)	The absolute path name indicator: **Y** The Absolute Path Name field contains an absolute path name for the key ring file. **N** The Absolute Path Name field does not contain an absolute path name for the key ring file.
	2457	2843	Relative File ID[2]	Char(16)	The relative file ID of the absolute path name.
	2473	2859	Absolute Path Name[1]	Char(5002)	The absolute path name of the key ring file.
	7475	7861	Object File ID	Char(16)	The file ID of the source or target file.
	7491	7877	ASP Name	Char(10)	Source or target file ASP name
	7501	7887	ASP Number	Char(5)	Source or target file ASP number
	7506	7892	Path Name CCSID	Binary(5)	The coded character set identifier for the absolute path name.
	7510	7896	Path name Country or Region ID	Char(2)	The Country or Region ID for the absolute path name

Table 168. KF (Key Ring File) Journal Entries (continued). QASYKFJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	7512	7898	Path Name Language ID	Char(3)	The language ID for the absolute path name.
	7515	7901	Path Name Length	Binary(4)	The length of the absolute path name.
	7517	7903	Path Name Indicator	Char(1)	The absolute path name indicator: **Y** The Absolute Path Name field contains an absolute path name for the source or target file. **N** The Absolute Path Name field does not contain an absolute path name for the source or target file.
	7518	7904	Relative File ID[3]	Char(16)	The relative file ID of the absolute path name.
	7534	7920	Absolute Path Name[1]	Char(5002)	The absolute path name of the source or target file.

1 This is a variable length field. The first 2 bytes contain the length of the path name.

2 When the path name indicator (offset 2456) is ″N″, this field will contain the relative file ID of the absolute path name at offset 2473. When the path name indicator is ″Y″, this field will contain 16 bytes of hex zeroes.

3 When the path name indicator (offset 7517) is ″N″, this field will contain the relative file ID of the absolute path name at offset 7534. When the path name indicator is ″Y″, this field will contain 16 bytes of hex zeroes.

4 The field will be blanks when it is not a certificate operation.

5 The field will be blanks when it is not a key ring file operation.

6 The field will be blanks when it is not a trusted root operation.

Table 169. LD (Link, Unlink, Search Directory) Journal Entries. QASYLDJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **L** Link directory **U** Unlink directory **K** Search directory
157			(Reserved area)	Char(20)	
	225	611	(Reserved area)	Char(18)	
	243	629	Object Name Length [1]	Binary (4)	The length of the object name.
177	245	631	Object Name CCSID[1]	Binary(5)	The coded character set identifier for the object name.

Table 169. LD (Link, Unlink, Search Directory) Journal Entries (continued). QASYLDJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
181	249	635	Object Name Country or Region ID[1]	Char(2)	The Country or Region ID for the object name.
183	251	637	Object Name Language ID[1]	Char(3)	The language ID for the object name.
186	254	640	(Reserved area)	Char(3)	
189	257	643	Parent File ID[1,2]	Char(16)	The file ID of the parent directory.
205	273	659	Object File ID[1,2]	Char(16)	The file ID of the object.
221	289	675	Object Name[1]	Char(512)	The name of the object.
	801	1187	Object File ID	Char(16)	The file ID of the object.
	817	1203	ASP Name	Char(10)	The name of the ASP device.
	827	1213	ASP Number	Char(5)	The number of the ASP device.
	832	1218	Path Name CCSID	Binary(5)	The coded character set identifier for the absolute path name.
	836	1222	Path Name Country or Region ID	Char(2)	The Country or Region ID for the absolute path name
	838	1224	Path Name Language ID	Char(3)	The language ID for the absolute path name.
	841	1227	Path Name Length	Binary(4)	The length of the absolute path name.
	843	1229	Path Name Indicator	Char(1)	The absolute path name indicator: **Y** The Absolute Path Name field contains an absolute path name for the object. **N** The Absolute Path Name field does not contain an absolute path name for the object.
	844	1230	Relative File ID[1]	Char(16)	The relative file ID of the absolute path name.
	860	1246	Absolute Path Name[2]	Char(5002)	The absolute path name of the object.

1 When the path name indicator (offset 843) is ″N″, this field will contain the relative file ID of the absolute path name. When the path name indicator is ″Y″, this field will contain 16 bytes of hex zeroes.

2 This is a variable length field. The first 2 bytes contain the length of the path name.

Table 170. ML (Mail Actions) Journal Entries. QASYMLJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **O** Mail log opened
157	225	611	User Profile	Char(10)	User profile name.

Table 170. ML (Mail Actions) Journal Entries (continued). QASYMLJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
167	235	621	User ID	Char(8)	User identifier
175	243	629	Address	Char(8)	User address

Table 171. NA (Attribute Change) Journal Entries. QASYNAJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **A** Change to network attribute. **T** Change to TCP/IP attribute.
157	225	611	Attribute	Char(10)	The name of the attribute.
167	235	621	New Attribute Value	Char(250)	The value of the attribute after it was changed.
417	485	871	Old Attribute Value	Char(250)	The value of the attribute before it was changed.

Table 172. ND (APPN Directory Search Filter) Journal Entries. QASYNDJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **A** Directory search filter violation
157	225	611	Filtered control point name	Char(8)	Filtered control point name
165	233	619	Filtered control point NETID.	Char(8)	Filtered control point NETID.
173	241	627	Filtered CP location name	Char(8)	Filtered CP location name.
181	249	635	Filtered CP location NETID	Char(8)	Filtered CP location NETID.
189	257	643	Partner location name	Char(8)	Partner location name.
197	265	651	Partner location NETID	Char(8)	Partner location NETID.
205	273	659	Inbound session	Char(1)	Inbound session. **Y** This is an inbound session **N** This is not an inbound session

Table 172. ND (APPN Directory Search Filter) Journal Entries (continued). QASYNDJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
206	274	660	Outbound session	Char(1)	Outbound session. **Y** This is an outbound session **N** This is not an outbound session

For more information about APPN Directory Search Filter and APPN End point, see the Information Center (see "Prerequisite and related information" on page xvi for details).

Table 173. NE (APPN End Point Filter) Journal Entries. QASYNEJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **A** End point filter violation
157	225	611	Local location name	Char(8)	Local location name.
165	233	619	Remote location name	Char(8)	Remote location name.
173	241	627	Remote NETID	Char(8)	Remote NETID.
181	249	635	Inbound session	Char(1)	Inbound session. **Y** This is an inbound session **N** This is not an inbound session
182	250	636	Outbound session	Char(1)	Outbound session. **Y** This is an outbound session **N** This is not an outbound session

For more information about APPN Directory Search Filter and APPN End point, see the Information Center (see "Prerequisite and related information" on page xvi for details).

Table 174. OM (Object Management Change) Journal Entries. QASYOMJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **M** Object moved to a different library. **R** Object renamed.
157	225	611	Old Object Name	Char(10)	The old name of the object.

Table 174. OM (Object Management Change) Journal Entries (continued). QASYOMJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
167	235	621	Old Library Name	Char(10)	The name of the library the old object is in.
177	245	631	Object Type	Char(8)	The type of object.
185	253	639	New Object Name	Char(10)	The new name of the object.
195	263	649	New Library Name	Char(10)	The name of the library the object was moved to.
205	273	659	(Reserved Area)	Char(20)	
225	293	679	Office User	Char(10)	The name of the office user.
235	303	689	Old Folder or Document Name	Char(12)	The old name of the folder or document.
247	315	701	(Reserved Area)	Char(8)	
255	323	709	Old Folder Path	Char(63)	The old path of the folder.
318	386	772	New Folder or Document Name	Char(12)	The new name of the folder or document.
330	398	784	(Reserved Area)	Char(8)	
338	406	792	New Folder Path	Char(63)	The new path of the folder.
401	469	855	Office on Behalf of User	Char(10)	User working on behalf of another user.
411			(Reserved Area)	Char(20)	
	479	865	(Reserved Area)	Char (18)	
	497	883	Object Name Length	Binary (4)	The length of the old object name field.
431	499	885	Object Name CCSID[1]	Binary(5)	The coded character set identifier for the object name.
435	503	889	Object Name Country or Region ID[1]	Char(2)	The Country or Region ID for the object name.
437	505	891	Object Name Language ID[1]	Char(3)	The language ID for the object name.
440	508	894	(Reserved area)	Char(3)	
443	511	897	Old Parent File ID[1,2]	Char(16)	The file ID of the old parent directory.
459	527	913	Old Object File ID[1,2]	Char(16)	The file ID of the old object.
475	543	929	Old Object Name[1]	Char(512)	The name of the old object.
987	1055	1441	New Parent File ID[1,2]	Char(16)	The file ID of the new parent directory.
1003	1071	1457	New Object Name[1, 2, 6]	Char(512)	The new name of the object.
	1583	1969	Object File ID[1,2]	Char(16)	The file ID of the object.
	1599	1985	ASP Name[7]	Char(10)	The name of the ASP device.

Table 174. OM (Object Management Change) Journal Entries (continued). QASYOMJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	1609	1995	ASP Number[7]	Char(5)	The number of the ASP device.
	1614	2000	Path Name CCSID	Binary(5)	The coded character set identifier for the absolute path name.
	1618	2004	Path Name Country or Region ID	Char(2)	The Country or Region ID for the absolute path name
	1620	2006	Path Name Language ID	Char(3)	The language ID for the absolute path name.
	1623	2009	Path Name Length	Binary(4)	The length of the absolute path name.
	1625	2011	Path Name Indicator	Char(1)	The absolute path name indicator: **Y** The Absolute Path Name field contains an absolute path name for the object. **N** The Absolute Path Name field does not contain an absolute path name for the object.
	1626	2012	Relative File ID[3]	Char(16)	The relative file ID of the absolute path name.
	1642	2028	Absolute Path Name[5]	Char(5002)	The old absolute path name of the object.
	6644	7030	Object File ID	Char(16)	The file ID of the object.
	6660	7046	ASP Name[8]	Char(10)	The name of the ASP device.
	6670	7056	ASP Number[8]	Char(5)	The number of the ASP device.
	6675	7061	Path Name CCSID	Binary(5)	The coded character set identifier for the absolute path name.
	6679	7065	Path Name Country or Region ID	Char(2)	The Country or Region ID for the absolute path name
	6681	7067	Path Name Language ID	Char(3)	The language ID for the absolute path name.
	6684	7070	Path Name Length	Binary(4)	The length of the absolute path name.
	6686	7072	Path Name Indicator	Char(1)	The absolute path name indicator: **Y** The Absolute Path Name field contains an absolute path name for the object. **N** The Absolute Path Name field does not contain an absolute path name for the object.
	6687	7073	Relative File ID[4]	Char(16)	The relative file ID of the absolute path name.
	6703	7089	Absolute Path Name[5]	Char(5002)	The new absolute path name of the object.

Table 174. OM (Object Management Change) Journal Entries (continued). QASYOMJE/J4/J5 Field Description File

Offset					
JE	J4	J5	Field	Format	Description

[1] These fields are used only for objects in the QOpenSys, "root" file systems, and user-defined file systems.

[2] An ID that has the left-most bit set and the rest of the bits zero indicates that the ID is NOT set.

[3] When the path name indicator (offset 1625) is ″N″, this field will contain the relative file ID of the absolute path name at offset 1642. When the path name indicator is ″Y″, this field will contain 16 bytes of hex zeroes.

[4] When the path name indicator (offset 6686) is ″N″, this field will contain the relative file ID of the absolute path name at offset 6703. When the path name indicator is ″Y″, this field will contain 16 bytes of hex zeroes.

[5] This is a variable length field. The first 2 bytes contain the length of the path name.

[6] There is no associated length field for this value. The string is null padded unless it is the full 512 characters long.

[7] If the old object is in a library, this is the ASP information of the object's library. If the old object is not in a library, this is the ASP information of the object.

[8] If the new object is in a library, this is the ASP information of the object's library. If the new object is not in a library, this is the ASP information of the object.

Table 175. OR (Object Restore) Journal Entries. QASYORJE/J4/J5 Field Description File

Offset					
JE	J4	J5	Field	Format	Description
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	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	225	611	Restored Object Name	Char(10)	The name of the restored object.
167	235	621	Restored Library Name	Char(10)	The name of the library of the restored object.
177	245	631	Object Type.	Char(8)	The type of object.
185	253	639	Save Object Name	Char(10)	The name of the save object.
195	263	649	Save Library Name	Char(10)	The name of the library from which the object was saved.
205	273	659	Program State[1]	Char(1)	**I** An inherit state program was restored. **Y** A system state program was restored. **N** A user state program was restored.
206	274	660	System Command [2]	Char(1)	**Y** A system command was restored. **N** A user state command was restored.
207			(Reserved Area)	Char(18)	

Table 175. OR (Object Restore) Journal Entries (continued). QASYORJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	275	661	SETUID Mode	Char(1)	The SETUID mode indicator.
					Y The SETUID mode bit for the restored object is on.
					N The SETUID mode bit for the restored object is not on.
	276	662	SETGID Mode	Char(1)	The SETGID mode indicator.
					Y The SETGID mode bit for the restored object is on.
					N The SETGID mode bit for the restored object is not on.
	277	663	Signature Status	Char(1)	The signature status of the restored object.
					B Signature was not in OS/400 format
					E Signature exists but is not verified
					F Signature does not match object content
					I Signature ignored
					N Unsignable object
					U Object unsigned
					S Signature is valid
	278	664	Reserved	Char(15)	
225	293	679	Office User	Char(10)	The name of the office user.
235	303	689	Restore DLO Name	Char(12)	The document library object name of the restored object.
247	315	701	(Reserved Area)	Char(8)	
255	323	709	Restore Folder Path	Char(63)	The folder into which the DLO was restored.
318	386	772	Save DLO Name	Char(12)	The DLO name of the saved object.
330	398	784	(Reserved Area)	Char(8)	
338	406	792	Save Folder Path	Char(63)	The folder from which the DLO was saved.
401	469	855	Office on Behalf of User	Char(10)	User working on behalf of another user.
411			(Reserved Area)	Char(20)	
	479	865	(Reserved Area)	Char(18)	
	497	883	Object Name Length	Binary (4)	The length of the Old Object Name field.
431	499	885	Object Name CCSID[3]	Binary(5)	The coded character set identifier for the object name.
435	503	889	Object Name Country or Region ID[3]	Char(2)	The Country or Region ID for the object name.
437	505	891	Object Name Language ID[3]	Char(3)	The language ID for the object name.

Table 175. OR (Object Restore) Journal Entries (continued). QASYORJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
440	508	894	(Reserved area)	Char(3)	
443	511	897	Parent File ID[3,4]	Char(16)	The file ID of the parent directory.
459	527	913	Object File ID[3,4]	Char(16)	The file ID of the object.
475	543	929	Object Name[3]	Char(512)	The name of the object.
	1055	1441	Old File ID	Char(16)	The file ID for the old object.
	1071	1457	Media File ID	Char(16)	The file ID (FID) that was stored on the media file. **Note:** The FID stored on the media is the FID the object had on the source system.
	1087	1473	Object File ID	Char(16)	The file ID of the object.
	1103	1489	ASP Name[7]	Char(10)	The name of the ASP device.
	1113	1499	ASP Number[7]	Char(5)	The number of the ASP device.
	1118	1504	Path Name CCSID	Binary(5)	The coded character set identifier for the absolute path name.
	1122	1508	Path Name Country or Region ID	Char(2)	The Country or Region ID for the absolute path name
	1124	1510	Path Name Language ID	Char(3)	The language ID for the absolute path name.
	1127	1513	Path Name Length	Binary(4)	The length of the absolute path name.
	1129	1515	Path Name Indicator	Char(1)	The absolute path name indicator: **Y** The Absolute Path Name field contains an absolute path name for the object. **N** The Absolute Path Name field does not contain an absolute path name for the object.
	1130	1516	Relative File ID[5]	Char(16)	The relative file ID of the absolute path name.
	1146	1532	Absolute Path Name[6]	Char(5002)	The absolute path name of the object.

[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.

[3] These fields are used only for objects in the QOpenSys file system and the "root" file system.

[4] An ID that has the left-most bit set and the rest of the bits zero indicates that the ID is NOT set.

[5] When the path name indicator (offset 1129) is ″N″, this field will contain the relative file ID of the absolute path name. When the path name indicator is ″Y″, this field will contain 16 bytes of hex zeroes.

[6] This is a variable length field. The first 2 bytes contain the length of the path name.

[7] If the object is in a library, this is the ASP information of the object's library. If the object is not in a library, this is the ASP information of the object.

Table 176. OW (Ownership Change) Journal Entries. QASYOWJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry.
					A Change of object owner
157	225	611	Object Name	Char(10)	The name of the object.
167	235	621	Library Name	Char(10)	The name of the library the object is in.
177	245	631	Object Type	Char(8)	The type of object.
185	253	639	Old Owner	Char(10)	Old owner of the object.
195	263	649	New Owner	Char(10)	New owner of the object.
205	273	659	(Reserved Area)	Char(20)	
225	293	679	Office User	Char(10)	The name of the office user.
235	303	689	DLO Name	Char(12)	The name of the document library object.
247	315	701	(Reserved Area)	Char(8)	
255	323	709	Folder Path	Char(63)	The path of the folder.
318	386	772	Office on Behalf of User	Char(10)	User working on behalf of another user.
328			(Reserved Area)	Char(20)	
	396	782	(Reserved Area)	Char(18)	
	414	800	Object Name Length	Binary (4)	The length of the new object name.
348	416	802	Object Name CCSID[1]	Binary(5)	The coded character set identifier for the object name.
352	420	806	Object Name Country or Region ID[1]	Char(2)	The Country or Region ID for the object name.
354	422	808	Object Name Language ID[1]	Char(3)	The language ID for the object name.
357	425	811	(Reserved area)	Char(3)	
360	428	814	Parent File ID[1,2]	Char(16)	The file ID of the parent directory.
376	444	830	Object File ID[1,2]	Char(16)	The file ID of the object.
392	460	846	Object Name[1]	Char(512)	The name of the object.
	972	1358	Object File ID	Char(16)	The file ID of the object.
	988	1374	ASP Name[5]	Char(10)	The name of the ASP device.
	998	1384	ASP Number[5]	Char(5)	The number of the ASP device.
	1003	1389	Path Name CCSID	Binary(5)	The coded character set identifier for the absolute path name.
	1007	1393	Path Name Country or Region ID	Char(2)	The Country or Region ID for the absolute path name
	1009	1395	Path Name Language ID	Char(3)	The language ID for the absolute path name.
	1012	1398	Path Name Length	Binary(4)	The length of the absolute path name.

Table 176. OW (Ownership Change) Journal Entries (continued). QASYOWJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	1014	1400	Path Name Indicator	Char(1)	The absolute path name indicator: **Y** The Absolute Path Name field contains an absolute path name for the object. **N** The Absolute Path Name field does not contain an absolute path name for the object.
	1015	1401	Relative File ID[3]	Char(16)	The relative file ID of the absolute path name.
	1031	1417	Absolute Path Name[4]	Char(5002)	The absolute path name of the object.

[1] These fields are used only for objects in the QOpenSys file system and the "root" file system.

[2] An ID that has the left-most bit set and the rest of the bits zero indicates that the ID is NOT set.

[3] When the path name indicator (offset 1014) is ″N″, this field will contain the relative file ID of the absolute path name. When the path name indicator is ″Y″, this field will contain 16 bytes of hex zeroes.

[4] This is a variable length field. The first 2 bytes contain the length of the path name.

[5] If the object is in a library, this is the ASP information of the object's library. If the object is not in a library, this is the ASP information of the object.

Table 177. O1 (Optical Access) Journal Entries. QASY01JE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	R-Read U-Update D-Delete C-Create Dir X-Release Held File
157	225	611	Object Type	Char(1)	F-File D-Directory End
158	226	612	Access Type	Char(1)	D-File Data A-File Directory Attributes
159	227	613	Device Name	Char(10)	Library LUD name
169	237	623	CSI Name	Char(8)	Side Object Name
177	245	631	CSI Library	Char(10)	Side Object Library
187	255	641	Volume Name	Char(32)	Optical volume name
219	287	673	Object Name	Char(256)	Optical directory/file name
		929	ASP name	Char(10)	ASP name for CSI library
		939	ASP number	Char(5)	ASP number for CSI library

Table 177. O1 (Optical Access) Journal Entries (continued). QASY01JE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
Note: This entry is used to audit the following optical functions: Open File or Directory Create Directory Delete File Directory Change or Retrieve Attributes Release Held Optical File					

Table 178. O2 (Optical Access) Journal Entries. QASY02JE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	C-Copy R-Rename B-Backup Dir or File S-Save Held File M-Move File
157	225	611	Object Type	Char(1)	F-File D-Directory
158	226	612	Src Device Name	Char(10)	Source library LUD name
168	236	622	Src CSI Name	Char(8)	Source Side Object Name
176	244	630	Src CSI Library	Char(10)	Source Side Object Library
186	254	640	Src Volume Name	Char(32)	Source Optical volume name
218	286	672	Src Obj Name	Char(256)	Source Optical directory/file name
474	542	928	Tgt Device Name	Char(10)	Target library LUD name
484	552	938	Tgt CSI Name	Char(8)	Target Side Object Name
492	560	946	Tgt CSI Library	Char(10)	Target Side Object Library
502	570	956	Tgt Volume Name	Char(32)	Target Optical volume name
534	602	988	Tgt Obj Name	Char(256)	Target Optical directory/file name
		1244	ASP name	Char(10)	ASP name for source CSI library
		1254	ASP number	Char(5)	ASP number for source CSI library
		1259	ASP name for target CSI library	Char(10)	ASP name for target CSI library
		1269	ASP number for target CSI library	Char(5)	ASP number for target CSI library

Table 179. O3 (Optical Access) Journal Entries. QASY03JE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	I-Initialize
					N-Rename
					B-Backup Volume
					C-Convert Backup Volume to Primary
					M-Import
					E-Export
					L-Change Auth. List
					A-Change Volume Attributes
					R-Absolute Read
157	225	611	Device Name	Char(10)	Library LUD name
167	235	621	CSI Name	Char(8)	Side Object Name
175	243	629	CSI Library	Char(10)	Side Object Library
185	253	639	Old Volume Name	Char(32)	Old Optical voulume name
217	285	671	New Volume Name [1]	Char(32)	New Optical volume name
249	317	703	Old Auth List [2]	Char(10)	Old Authorization List
259	327	713	New Auth List [3]	Char(10)	New Authorization List
269	337	723	Address [4]	Binary(5)	Starting Block
273	341	727	Length [4]	Binary(5)	Length read
		731	ASP name	Char(10)	ASP name for CSI library
		741	ASP number	Char(5)	ASP number for CSI library

[1] This field contains the new volume name for Initialize, Rename, and Convert functions; it contains the backup bolume name for Backup functions. It contains volume name for Import, Export, Change Authorization List, Change Volume Attributes, and Sector Read.

[2] Used for Import, Export, and Change Authorization List only.

[3] Used for Change Authorization List only.

[4] Used for Sector Read only.

Table 180. PA (Program Adopt) Journal Entries. QASYPAJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.

Table 180. PA (Program Adopt) Journal Entries (continued). QASYPAJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
156	224	610	Entry Type	Char(1)	The type of entry.
					A Change program to adopt owner's authority.
					J Java program adopts owner's authority.
					M Modify object's SETUID or SETGID mode indicator.
157	225	611	Program Name[3]	Char(10)	The name of the program.
167	235	621	Program Library[3]	Char(10)	The name of the library where the program is found.
177	245	631	Object Type	Char(8)	The type of object.
185	253	639	Owner	Char(10)	The name of the owner.
	263	649	Reserved	Char(18)	
	281	667	Object Name Length[1]	Binary (4)	The length of the object name.
	283	669	Object Name CCSID[1]	Binary(5)	The coded character set identifier for the object name.
	287	673	Object Name Country or Region ID	Char(2)	The Country or Region ID for the object name.
	289	675	Object Name Language ID[1]	Char(3)	The language ID for the object name.
	292	678	Reserved	Char(3)	
	295	681	Parent ID[1, 2, 3]	Char(16)	Parent File ID.
	311	697	Object File ID [3]	Char(16)	File ID for the object
	327	713	Object Name[1]	Char(512)	Object name for the object.
	839	1225	SETUID Mode	Char(1)	The SETUID mode indicator.
					Y The SETUID mode bit is on for the object.
					N The SETUID mode bit is not on for the object.
	840	1226	SETGID Mode	Char(1)	The SETGID mode indicator.
					Y The SETGID mode bit is on for the object.
					N The SETGID mode bit is not on for the object.
	841	1227	Primary Group Owner	Char(10)	The name of the primary group owner.
	851	1237	Object File ID	Char(16)	The file ID of the object.
	867	1253	ASP Name[6]	Char(10)	The name of the ASP device.
	877	1263	ASP Number[6]	Char(5)	The number of the ASP device.
	882	1268	Path Name CCSID	Binary(5)	The coded character set identifier for the absolute path name.
	886	1272	Path Name Country or Region ID	Char(2)	The Country or Region ID for the absolute path name
	888	1274	Path Name Language ID	Char(3)	The language ID for the absolute path name.
	891	1277	Path Name Length	Binary(4)	The length of the absolute path name.

Table 180. PA (Program Adopt) Journal Entries (continued). QASYPAJE/J4/J5 Field Description File

Offset					
JE	J4	J5	Field	Format	Description
	893	1279	Path Name Indicator	Char(1)	The absolute path name indicator: **Y** The Absolute Path Name field contains an absolute path name for the object. **N** The Absolute Path Name field does not contain an absolute path name for the object.
	894	1280	Relative File ID[4]	Char(16)	The relative file ID of the absolute path name.
	910	1296	Absolute Path Name[5]	Char(5002)	The absolute path name of the object.

[1] These fields are used only for objects in the QOpenSys and "root" file systems.

[2] An ID that has the left-most bit set and the rest of the bits zero indicates that the ID is NOT set.

[3] When the entry type is ″J″, the program name and the library name fields will contain ″*N″. In addition, the parent file ID and the object file ID fields will contain binary zeroes.

[4] When the path name indicator (offset 893) is ″N″, this field will contain the relative file ID of the absolute path name. When the path name indicator is ″Y″, this field will contain 16 bytes of hex zeroes.

[5] This is a variable length field. The first 2 bytes contain the length of the path name.

[6] If the object is in a library, this is the ASP information of the object's library. If the object is not in a library, this is the ASP information of the object.

Table 181. PG (Primary Group Change) Journal Entries. QASYPGJE/J4/J5 Field Description File

Offset					
JE	J4	J5	Field	Format	Description
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **A** Change primary group.
157	225	611	Object Name	Char(10)	The name of the object.
167	235	621	Object Library	Char(10)	The name of the library where the object is found.
177	245	631	Object Type	Char(8)	The type of object.
185	253	639	Old Primary Group	Char(10)	The previous primary group for the object.[5]
195	263	649	New Primary Group	Char(10)	The new primary group for the object.
					Authorities for new primary group:
205	273	659	Object Existence	Char(1)	**Y** *OBJEXIST
206	274	660	Object Management	Char(1)	**Y** *OBJMGT
207	275	661	Object Operational	Char(1)	**Y** *OBJOPR
208	276	662	Object Alter	Char(1)	**Y** *OBJALTER
209	277	663	Object Reference	Char(1)	**Y** *OBJREF
210	278	664	(Reserved Area)	Char(10)	

Table 181. PG (Primary Group Change) Journal Entries (continued). QASYPGJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
220	288	674	Authorization List Management	Char(1)	**Y** *AUTLMGT
221	289	675	Read Authority	Char(1)	**Y** *READ
222	290	676	Add Authority	Char(1)	**Y** *ADD
223	291	677	Update Authority	Char(1)	**Y** *UPD
224	292	678	Delete Authority	Char(1)	**Y** *DLT
225	293	679	Execute Authority	Char(1)	**Y** *EXECUTE
226	294	680	(Reserved Area)	Char(10)	
236	304	690	Exclude Authority	Char(1)	**Y** *EXCLUDE
237	305	691	Revoke Old Primary Group	Char(1)	**Y** Revoke authority for previous primary group. **' '** Do not revoke authority for previous primary group.
238	306	692	(Reserved Area)	Char (20)	
258	326	712	Office User	Char(10)	The name of the office user.
268	336	722	DLO Name	Char(12)	The name of the document library object or folder.
280	348	734	(Reserved Area)	Char(8)	
288	356	742	Folder Path	Char(63)	The path of the folder.
351	419	805	Office on Behalf of User	Char(10)	User working on behalf of another user.
361			(Reserved Area)	Char(20)	
	429	815	(Reserved Area)	Char(18)	
	447	833	Object Name Length[1]	Binary (4)	The length of the object name.
381	449	835	Object Name CCSID[1]	Binary(5)	The coded character set identifier for the object name.
385	453	839	Object Name Country or Region ID[1]	Char(2)	The Country or Region ID for the object name.
387	455	841	Object Name Language ID[1]	Char(3)	The language ID for the object name.
390	458	844	(Reserved area)	Char(3)	
393	461	847	Parent File ID[1,2]	Char(16)	The file ID of the parent directory.
409	477	863	Object File ID[1,2]	Char(16)	The file ID of the object.
425	493	879	Object Name[1]	Char(512)	The name of the object.
	1005	1391	Object File ID	Char(16)	The file ID of the object.
		1407	ASP Name[6]	Char(10)	The name of the ASP device.
		1417	ASP Number[6]	Char(5)	The number of the ASP device.
	1035	1422	Path Name CCSID	Binary(5)	The coded character set identifier for the absolute path name.
	1040	1426	Path Name Country or Region ID	Char(2)	The Country or Region ID for the absolute path name

Table 181. PG (Primary Group Change) Journal Entries (continued). QASYPGJE/J4/J5 Field Description File

Offset					
JE	J4	J5	Field	Format	Description
	1042	1428	Path Name Language ID	Char(3)	The language ID for the absolute path name.
	1045	1431	Path Name Length	Binary(4)	The length of the absolute path name.
	1047	1433	Path Name Indicator	Char(1)	The absolute path name indicator: **Y** The Absolute Path Name field contains an absolute path name for the object. **N** The Absolute Path Name field does not contain an absolute path name for the object.
	1048	1434	Relative File ID[3]	Char(16)	The relative file ID of the absolute path name.
	1064	1450	Absolute Path Name[4]	Char(5002)	The absolute path name of the object.

[1] These fields are used only for objects in the QOpenSys and "root" file systems.

[2] An ID that has the left-most bit set and the rest of the bits zero indicates that the ID is NOT set.

[3] When the path name indicator (offset 1047) is ″N″, this field will contain the relative file ID of the absolute path name. When the path name indicator is ″Y″, this field will contain 16 bytes of hex zeroes.

[4] This is a variable length field. The first 2 bytes contain the length of the path name.

[5] A value of *N implies the value of the Old Primary Group was not available.

[6] If the object is in a library, this is the ASP information of the object's library. If the object is not in a library, this is the ASP information of the object.

Table 182. PO (Printer Output) Journal Entries. QASYPOJE/J4/J5 Field Description File

Offset					
JE	`J4	J5	Field	Format	Description
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Output Type	Char(1)	The type of output. **D** Direct print **R** Sent to remote system for printing **S** Spooled file printed
157	225	611	Status After Printing	Char(1)	**D** Deleted after printed **H** Held after printed **S** Saved after printed **' '** Direct print
158	226	612	Job Name	Char(10)	The first part of the qualified job name.
168	236	622	Job User Name	Char(10)	The second part of the qualified job name.
178	246	632	Job Number	Zoned(6,0)	The third part of the qualified job name.
184	252	638	User Profile	Char(10)	The user profile that created the output.
194	262	648	Output Queue	Char(10)	The output queue containing the spooled file.[1]
204	272	658	Output Queue Library Name	Char(10)	The name of the library containing the output queue.[1]

Table 182. PO (Printer Output) Journal Entries (continued). QASYPOJE/J4/J5 Field Description File

Offset					
JE	**`J4**	**J5**	**Field**	**Format**	**Description**
214	282	668	Device Name	Char(10)	The device where the output was printed[2].
224	292	678	Device Type	Char(4)	The type of printer device[2].
228	296	682	Device Model	Char(4)	The model of the printer device[2].
232	300	686	Device File Name	Char(10)	The name of the device file used to access the printer.
242	310	696	Device File Library	Char(10)	The name of the library for the device file.
252	320	706	Spooled File Name	Char(10)	The name of the spooled file [1]
262	330	716	Short Spooled File Number	Char(4)	The number of the spooled file [1]. Set to blank if too long.
266	334	720	Form Type	Char(10)	The form type of the spooled file.
276	344	730	User Data	Char(10)	The user data associated with the spooled file [1].
286			(Reserved area)	Char(20)	
	354	740	Spooled File Number	Char(6)	The number of the spooled file.
	360	746	Reserved Area	Char(14)	
306	374	760	Remote System	Char(255)	Name of the remote system to which printing was sent.
561	629	1015	Remote System Print Queue	Char(128)	The name of the output queue on the remote system.
	757	1143	Spooled File Job system Name	Char (8)	The name of the system on which the spooled file resides.
	765	1151	Spooled File Create Date	Char (7)	The spooled file create date (CYYMMDD)
	772	1158	Spooled File Create Time	Char(6)	The spooled file create time (HHMMSS).
		1164	ASP Name	Char(10)	ASP name for the device library
		1174	ASP number	Char(5)	ASP number for device file library

1 This field is blank if the type of output is direct print.

2 This field is blank if the type of output is remote print.

Table 183. PS (Profile Swap) Journal Entries. QASYPSJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.

Table 183. PS (Profile Swap) Journal Entries (continued). QASYPSJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
156	224	610	Entry Type	Char(1)	The type of entry. **A** Profile swap during pass-through. **E** End work on behalf of relationship. **H** Profile handle generated by the QSYGETPH API. **I** All profile tokens were invalidated **M** Maximum number of profile tokens have been generated. **P** Profile token generated for user. **R** All profile tokens for a user have been removed. **S** Start work on behalf of relationship **V** User profile authenticated
157	225	611	User Profile	Char(10)	User profile name.
167	235	621	Source Location	Char(8)	Pass-through source location.
175	243	629	Original Target User Profile	Char(10)	Original pass-through target user profile.
185	253	639	New Target User Profile	Char(10)	New pass-through target user profile.
195	263	649	Office User	Char(10)	Office user starting or ending on behalf of relationship.
205	273	659	On Behalf of User	Char(10)	User on behalf of whom the office user is working.
215	283	669	Profile Token Type	Char(1)	The type of the profile token that was generated. **M** Multiple-use profile token **R** Multiple-use regenerated profile token **S** Single-use profile token
216	284	670	Profile Token Timeout	Binary(4)	The number of seconds the profile token is valid.

Table 184. PW (Password) Journal Entries. QASYPWJE/J4/J5 Field Description File

Offset					Description
JE	**J4**	**J5**	**Field**	**Format**	
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.

Table 184. PW (Password) Journal Entries (continued). QASYPWJE/J4/J5 Field Description File

Offset					Description
JE	**J4**	**J5**	**Field**	**Format**	
156	224	610	Violation Entry Type	Char(1)	The type of violation **A** APPC bind failure **D** Service tools user ID name not valid **E** Service tools user ID password not valid **P** Password not valid **U** User name not valid **X** Service tools user ID is disabled **Y** Service tools user ID not valid **Z** Service tools user ID password not valid
157	225	611	User Name	Char(10)	The job user name or the service tools user ID name.
167	235	621	Device name	Char(40)	The name of the device or communications device on which the password or user ID was entered. If the entry type is X, Y, or Z, this field will contain the name of the service tool being accessed.
207	275	661	Remote Location Name	Char(8)	Name of the remote location for the APPC bind.
215	283	669	Local Location Name	Char(8)	Name of the local location for the APPC bind.
223	291	677	Network ID	Char(8)	Network ID for the APPC bind.

Table 185. RA (Authority Change for Restored Object) Journal Entries. QASYRAJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **A** Changes to authority for object restored
157	225	611	Object Name	Char(10)	The name of the object.
167	235	621	Library Name	Char(10)	The name of the library the object is in.
177	245	631	Object Type	Char(8)	The type of object.
185	253	639	Authorization List Name	Char(10)	The name of the authorization list.
195	263	649	Public Authority	Char(1)	**Y** Public authority set to *EXCLUDE.
196	264	650	Private Authority	Char(1)	**Y** Private authority removed.
197	265	651	AUTL Removed	Char(1)	**Y** Authorization list removed from object.
198	266	652	(Reserved Area)	Char(20)	

Table 185. RA (Authority Change for Restored Object) Journal Entries (continued). QASYRAJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
218	286	672	DLO Name	Char(12)	The name of the document library object.
230	298	684	(Reserved Area)	Char(8)	
238	306	692	Folder Path	Char(63)	The folder containing the document library object.
301			(Reserved Area)	Char(20)	
	369	755	(Reserved Area)	Char(18)	
	387	773	Object Name Length	Binary(4)	The length of the object name.
321	389	775	Object Name CCSID[1]	Binary(5)	The coded character set identifier for the object name.
325	393	779	Object Name Country or Region ID[1]	Char(2)	The Country or Region ID for the object name.
327	395	781	Object Name Language ID[1]	Char(3)	The language ID for the object name.
330	398	784	(Reserved area)	Char(3)	
333	401	787	Parent File ID[1,2]	Char(16)	The file ID of the parent directory.
349	417	803	Object File ID[1,2]	Char(16)	The file ID of the object.
365	433	819	Object Name[1]	Char(512)	The name of the object.
	945	1331	Object File ID	Char(16)	The file ID of the object.
	961	1347	ASP Name[5]	Char(10)	The name of the ASP device.
	971	1357	ASP Number[5]	Char(5)	The number of the ASP device.
	976	1362	Path Name CCSID	Binary(5)	The coded character set identifier for the absolute path name.
	980	1366	Path Name Country or Region ID	Char(2)	The Country or Region ID for the absolute path name
	982	1368	Path Name Language ID	Char(3)	The language ID for the absolute path name.
	985	1371	Path Name Length	Binary(4)	The length of the absolute path name.
	987	1373	Path Name Indicator	Char(1)	The absolute path name indicator: **Y** The Absolute Path Name field contains an absolute path name for the object. **N** The Absolute Path Name field does not contain an absolute path name for the object.
	988	1374	Relative File ID[3]	Char(16)	The relative file ID of the absolute path name.
	1004	1390	Absolute Path Name[4]	Char(5002)	The absolute path name of the object.

Table 185. RA (Authority Change for Restored Object) Journal Entries (continued). QASYRAJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
[1]	These fields are used only for objects in the QOpenSys and "root" file systems.				
[2]	An ID that has the left-most bit set and the rest of the bits zero indicates that the ID is NOT set.				
[3]	When the path name indicator (offset 987) is ″N″, this field will contain the relative file ID of the absolute path name. When the path name indicator is ″Y″, this field will contain 16 bytes of hex zeroes.				
[4]	This is a variable length field. The first 2 bytes contain the length of the path name.				
[5]	If the object is in a library, this is the ASP information of the object's library. If the object is not in a library, this is the ASP information of the object.				

Table 186. RJ (Restoring Job Description) Journal Entries. QASYRJJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **A** Restoring a job description that had a user profile specified in the USER parameter.
157	225	611	Job Description Name	Char(10)	The name of the job description restored.
167	235	621	Library Name	Char(10)	The name of the library the job description was restored to.
177	245	631	Object Type	Char(8)	The type of object.
185	253	639	User Name	Char(10)	The name of the user profile specified in the job description.
		649	ASP name	Char(10)	ASP name for JOBD library
		659	ASP number	Char(5)	ASP number for JOBD library

Table 187. RO (Ownership Change for Restored Object) Journal Entries. QASYROJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **A** Restoring objects that had ownership changed when restored
157	225	611	Object Name	Char(10)	The name of the object.
167	235	621	Library Name	Char(10)	The name of the library the object is in.
177	245	631	Object Type	Char(8)	The type of object.
185	253	639	Old Owner	Char(10)	The name of the owner before ownership was changed.

Table 187. RO (Ownership Change for Restored Object) Journal Entries (continued). QASYROJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
195	263	649	New Owner	Char(10)	The name of the owner after ownership was changed.
205	273	659	(Reserved Area)	Char(20)	
225	293	679	DLO Name	Char(12)	The name of the document library object.
237	305	691	(Reserved Area)	Char(8)	
245	313	699	Folder Path	Char(63)	The folder into which the object was restored.
308			(Reserved Area)	Char(20)	
	376	762	(Reserved Area)	Char(18)	
	394	780	Object Name Length[1]	Binary(4)	The length of the object name.
328	396	782	Object Name CCSID[1]	Binary(5)	The coded character set identifier for the object name.
332	400	786	Object Name Country or Region ID[1]	Char(2)	The Country or Region ID for the object name.
334	402	788	Object Name Language ID[1]	Char(3)	The language ID for the object name.
337	405	791	(Reserved area)	Char(3)	
340	408	794	Parent File ID[1,2]	Char(16)	The file ID of the parent directory.
356	424	810	Object File ID[1,2]	Char(16)	The file ID of the object.
372	440	826	Object Name[1]	Char(512)	The name of the object.
	952	1338	Object File ID	Char(16)	The file ID of the object.
	968	1354	ASP Name[5]	Char(10)	The name of the ASP device.
	978	1364	ASP Number[5]	Char(5)	The number of the ASP device.
	983	1369	Path Name CCSID	Binary(5)	The coded character set identifier for the absolute path name.
	987	1373	Path Name Country or Region ID	Char(2)	The Country or Region ID for the absolute path name
	989	1375	Path Name Language ID	Char(3)	The language ID for the absolute path name.
	992	1378	Path Name Length	Binary(4)	The length of the absolute path name.
	994	1380	Path Name Indicator	Char(1)	The absolute path name indicator: **Y** The Absolute Path Name field contains an absolute path name for the object. **N** The Absolute Path Name field does not contain an absolute path name for the object.
	995	1381	Relative File ID[3]	Char(16)	The relative file ID of the absolute path name.
	1011	1397	Absolute Path Name[4]	Char(5002)	The absolute path name of the object.

Table 187. RO (Ownership Change for Restored Object) Journal Entries (continued). QASYROJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
[1]	These fields are used only for objects in the QOpenSys and "root" file systems.				
[2]	An ID that has the left-most bit set and the rest of the bits zero indicates that the ID is NOT set.				
[3]	When the path name indicator (offset 994) is ″N″, this field will contain the relative file ID of the absolute path name. When the path name indicator is ″Y″, this field will contain 16 bytes of hex zeroes.				
[4]	This is a variable length field. The first 2 bytes contain the length of the path name.				
[5]	If the object is in a library, this is the ASP information of the object's library. If the object is not in a library, this is the ASP information of the object.				

Table 188. RP (Restoring Programs that Adopt Authority) Journal Entries. QASYRPJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **A** Restoring programs that adopt the owner's authority
157	225	611	Program Name	Char(10)	The name of the program
167	235	621	Program Library	Char(10)	The name of the library in which the program is located
177	245	631	Object Type	Char(8)	The type of object
185	253	639	Owner Name	Char(10)	Name of the owner
	263	649	(Reserved Area)	Char(18)	
	281	667	Object Name Length[1]	Binary (4)	The length of the object name.
	283	669	Object Name CCSID[1]	Binary (5)	The coded character set identifier for the object name.
	287	673	Object Name Country or Region ID[1]	Char (2)	The Country or Region ID for the object name.
	289	675	Object name Language ID[1]	Char (3)	The language ID for the object name.
	292	678	(Reserved Area)	Char (3)	
	295	681	Parent File ID[1,2]	Char (16)	The file ID of the parent directory.
	311	697	Object File ID[1,2]	Char (16)	The file ID of the object.
	327	713	Object Name[1]	Char (512)	The name of the object.
	839	1225	Object File ID	Char(16)	The file ID of the object.
	855	1241	ASP Name[5]	Char(10)	The name of the ASP device.
	865	1251	ASP Number[5]	Char(5)	The number of the ASP device.
	870	1256	Path Name CCSID	Binary(5)	The coded character set identifier for the absolute path name.
	874	1260	Path Name Country or Region ID	Char(2)	The Country or Region ID for the absolute path name
	876	1262	Path Name Language ID	Char(3)	The language ID for the absolute path name.

Table 188. RP (Restoring Programs that Adopt Authority) Journal Entries (continued). QASYRPJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	879	1265	Path Name Length	Binary(4)	The length of the absolute path name.
	881	1267	Path Name Indicator	Char(1)	The absolute path name indicator: **Y** The Absolute Path Name field contains an absolute path name for the object. **N** The Absolute Path Name field does not contain an absolute path name for the object.
	882	1268	Relative File ID[3]	Char(16)	The relative file ID of the absolute path name.
	898	1284	Absolute Path Name[4]	Char(5002)	The absolute path name of the object.

[1] These fields are used only for objects in the QOpenSys and the 'root' file system.

[2] If an ID that has the left-most bit set and the rest of the bits are zero, the ID is **not** set.

[3] When the path name indicator (offset 994) is "N", this field will contain the relative file ID of the absolute path name. When the path name indicator is "Y", this field will contain 16 bytes of hex zeroes.

[4] This is a variable length field. The first 2 bytes contain the length of the path name.

[5] If the object is in a library, this is the ASP information of the object's library. If the object is not in a library, this is the ASP information of the object.

Table 189. RQ (Restoring Change Request Descriptor Object) Journal Entries. QASYRQJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **A** Restore *CRQD object that adopts authority.
157	225	611	Object Name	Char(10)	The name of the change request descriptor.
167	235	621	Object Library	Char(10)	The name of the library where the change request descriptor is found.
177	245	631	Object Type	Char(8)	The type of object.
		639	ASP name	Char(10)	ASP name for CRQD library
		649	ASP number	Char(5)	ASP number for CRQD library

Table 190. RU (Restore Authority for User Profile) Journal Entries. QASYRUJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **A** Restoring authority to user profiles

Table 190. RU (Restore Authority for User Profile) Journal Entries (continued). QASYRUJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
157	225	611	User Name	Char(10)	The name of the user profile whose authority was restored.
167	235	621	Library Name	Char(10)	The name of the library.
177	245	631	Object Type	Char(8)	The type of object.
	253	639	Authority Restored	Char(1)	Indicates whether all authorities were restored for the user. **A** All authorities were restored **S** Some authorities not restored

Table 191. RZ (Primary Group Change for Restored Object) Journal Entries. QASYRZJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **A** Primary group changed.
157	225	611	Object Name	Char(10)	The name of the object.
167	235	621	Object Library	Char(10)	The name of the library where the object is found.
177	245	631	Object Type	Char(8)	The type of object.
185	253	639	Old Primary Group	Char(10)	The previous primary group for the object.
195	263	649	New Primary Group	Char(10)	The new primary group for the object.
205	273	659	(Reserved Area)	Char(20)	
225	293	679	DLO Name	Char(12)	The name of the document library object.
237	305	691	(Reserved Area)	Char(8)	
245	313	699	Folder Path	Char(63)	The folder into which the object was restored.
308			(Reserved Area)	Char(20)	
	376	762	(Reserved Area)	Char(18)	
	394	780	Object Name Length[1]	Binary(4)	The length of the object name.
328	396	782	Object Name CCSID[1]	Binary(5)	The coded character set identifier for the object name.
332	400	786	Object Name Country or Region ID[1]	Char(2)	The Country or Region ID for the object name.
334	402	788	Object Name Language ID[1]	Char(3)	The language ID for the object name.
337	405	791	(Reserved area)	Char(3)	
340	408	794	Parent File ID[1,2]	Char(16)	The file ID of the parent directory.

Table 191. RZ (Primary Group Change for Restored Object) Journal Entries (continued). QASYRZJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
356	424	810	Object File ID[1,2]	Char(16)	The file ID of the object.
372	440	826	Object Name[1]	Char(512)	The name of the object.
	952	1338	Object File ID	Char(16)	The file ID of the object.
	968	1354	ASP Name	Char(10)	The name of the ASP device.
	978	1364	ASP Number	Char(5)	The number of the ASP device.
	983	1369	Path Name CCSID	Binary(5)	The coded character set identifier for the absolute path name.
	987	1373	Path Name Country or Region ID	Char(2)	The Country or Region ID for the absolute path name
	989	1375	Path Name Language ID	Char(3)	The language ID for the absolute path name.
	992	1378	Path Name Length	Binary(4)	The length of the absolute path name.
	994	1380	Path Name Indicator	Char(1)	The absolute path name indicator: **Y** The Absolute Path Name field contains an absolute path name for the object. **N** The Absolute Path Name field does not contain an absolute path name for the object.
	995	1381	Relative File ID[3]	Char(16)	The relative file ID of the absolute path name.
	1011	1397	Absolute Path Name[4]	Char(5002)	The absolute path name of the object.

[1] These fields are used only for objects in the QOpenSys and "root" file systems.

[2] An ID that has the left-most bit set and the rest of the bits zero indicates that the ID is NOT set.

[3] When the path name indicator (offset 1014) is ″N″, this field will contain the relative file ID of the absolute path name. When the path name indicator is ″Y″, this field will contain 16 bytes of hex zeroes.

[4] This is a variable length field. The first 2 bytes contain the length of the path name.

Table 192. SD (Change System Distribution Directory) Journal Entries. QASYSDJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **S** System directory change

Table 192. SD (Change System Distribution Directory) Journal Entries (continued). QASYSDJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
157	225	611	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	228	614	Type of record	Char(4)	**DIRE** Directory **DPTD** Department details **SHDW** Directory shadow **SRCH** Directory search
164	232	618	Originating System	Char(8)	The system originating the change
172	240	626	User Profile	Char(10)	The user profile making the change
182	250	636	Requesting system	Char(8)	The system requesting the change
190	258	644	Function Requested	Char(6)	**INIT** Initialization **OFFLIN** Offline initialization **REINIT** Reinitialization **SHADOW** Normal shadowing **STPSHD** Stop shadowing
196	264	650	User ID	Char(8)	The user ID being changed
204	272	658	Address	Char(8)	The address being changed
212	280	666	Network User ID	Char(47)	The network user ID being changed

Table 193. SE (Change of Subsystem Routing Entry) Journal Entries. QASYSEJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.

Table 193. SE (Change of Subsystem Routing Entry) Journal Entries (continued). QASYSEJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
156	224	610	Entry Type	Char(1)	The type of entry. **A** Subsystem routing entry changed
157	225	611	Subsystem Name	Char(10)	The name of the object
167	235	621	Library Name	Char(10)	The name of the library the object is in
177	245	631	Object Type	Char(8)	The type of object.
185	253	639	Program Name	Char(10)	The name of the program that changed the routing entry
195	263	649	Library Name	Char(10)	The name of the library for the program
205	273	659	Sequence Number	Char(4)	The sequence number
209	277	663	Command Name	Char(3)	The type of command used **ADD** ADDRTGE **CHG** CHGRTGE **RMV** RMVRTGE
		666	ASP name for SBSD library	Char(10)	ASP name for SBSD library
		676	ASP number for SBSD library	Char(5)	ASP number for SBSD library
		681	ASP name for program library	Char(10)	ASP name for program library
		691	ASP number for program library	Char(5)	ASP number for program library

Table 194. SF (Action to Spooled File) Journal Entries. QASYSFJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	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** Security-relevant spooled file changed. See footnote 2. **V** Only nonsecurity-relevant spooled file attributes changed.

Table 194. SF (Action to Spooled File) Journal Entries (continued). QASYSFJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
157	225	611	Database File Name	Char(10)	The name of the database file containing the spooled file
167	235	621	Library Name	Char(10)	The name of the library for the database file
177	245	631	Object Type	Char(8)	The object type of the database file
185	253	639	Reserved area	Char(10)	
195	263	649	Member Name	Char(10)	The name of the file member.
205	273	659	Spooled File Name	Char(10)	The name of the spooled file [1].
215	283	669	Short Spooled File Number	Char(4)	The number of the spooled file [1]. If the spooled file number is larger than 4 bytes, this field will be blank and the Spooled File Number field (offset 307) will be used.
219	287	673	Output Queue Name	Char(10)	The name of the output queue containing the spooled file.
229	297	683	Output Queue Library	Char(10)	The name of the library for the output queue.
239			Reserved area	Char(20)	
	307	693	Spooled File Number	Char(6)	The number of the spooled file.
	313	699	Reserved Area	Char(14)	
259	327	713	Old Copies	Char(3)	Number of old copies of the spooled file
262	330	716	New Copies	Char(3)	Number of new copies of the spooled file
265	333	719	Old Printer	Char(10)	Old printer for the spooled file
275	343	729	New Printer	Char(10)	New printer for the spooled file
285	353	739	New Output Queue	Char(10)	New output queue for the spooled file
295	363	749	New Output Queue Library	Char(10)	Library for the new output queue
305	373	759	Old Form Type	Char(10)	Old form type of the spooled file
315	383	769	New Form Type	Char(10)	New form type of the spooled file
325	393	779	Old Restart Page	Char(8)	Old restart page for the spooled file
333	401	787	New Restart Page	Char(8)	New restart page for the spooled file
341	409	795	Old Page Range Start	Char(8)	Old page range start of the spooled file
349	417	803	New Page Range Start	Char(8)	New page range start of the spooled file
357	425	811	Old Page Range End	Char(8)	Old page range end of the spooled file
365	433	819	New Page Range End	Char(8)	New page range end of the spooled file
	441	827	Spooled File Job Name	Char(10)	The name of the spooled file job.
	451	837	Spooled File Job User	Char(10)	The user for the spooled file job.
	461	847	Spooled File Job Number	Char(6)	The number for the spooled file job.
	467	853	Old Drawer	Char(8)	Old source drawer.
	475	861	New Drawer	Char(8)	New source drawer.

Table 194. SF (Action to Spooled File) Journal Entries (continued). QASYSFJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	483	869	Old Page Definition Name	Char(10)	Old page definition name.
	493	879	Old Page Definition Library	Char(10)	Old page definition library name.
	503	889	New Page Definition Name	Char(10)	New page definition name.
	513	899	New Page Definition Library	Char(10)	New page definition library.
	523	909	Old Form Definition Name	Char(10)	Old form definition name.
	533	919	Old Form Definition library	Char(10)	Old form definition library name.
	543	929	Name of new form definition	Char(10)	Name of new form definition
	553	939	New Form Definition Library	Char(10)	New form definition library name.
	563	949	Old User Defined Option 1	Char(10)	Old user-defined option 1.
	573	959	Old User Defined Option 2	Char(10)	Old user-defined option 2.
	583	969	Old User Defined Option 3	Char(10)	Old user-defined option 3.
	593	979	Old User Defined Option 4	Char(10)	Old user-defined option 4.
	603	989	New User Defined Option 1	Char(10)	New user-defined option 1.
	613	999	New User Defined Option 2	Char(10)	New user-defined option 2.
	623	1009	New User Defined Option 3	Char(10)	New user-defined option 3.
	633	1019	New User Defined Option 4	Char(10)	New user-defined option 4.
	643	1029	Old User Defined Object	Char(10)	Old user-defined object name.
	653	1039	Old User Defined Object Library	Char(10)	Old user-defined library name.

Table 194. SF (Action to Spooled File) Journal Entries (continued). QASYSFJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	663	1049	Old User Defined Object Type	Char(10)	Old user-defined object type.
	673	1059	New User Defined Object	Char(10)	New user-defined object.
	683	1069	New User Defined Object Library	Char(10)	New user-defined object library name.
	693	1079	New User Defined Object Type	Char(10)	New user-defined object type.
	703	1089	Spooled File Job System Name	Char(8)	The name of the system on which the spooled file resides.
	711	1097	Spooled File Create Date	Char(7)	The spooled file create date (CYYMMDD).
	718	1104	Spooled File Create Time	Char(6)	The spooled file create time (HHMMSS).
		1110	Name of old user defined data	Char(255)	Name of old user defined data
		1365	Name of new user defined data	Char(255)	Name of new user defined data

[1] This field is blank when the type of entry is I (inline print).

Table 195. SG (Asychronous Signals) Journal Entries. QASYSGJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	1	1			Heading fields common to all entry types. See Table 141 on page 501 and Table 142 on page 503 for field listing.
	224	610	Entry Type	Char(1)	The type of entry.
					A Asynchronous iSeries signal processed
					P Asynchronous Private Address Space Environment (PASE) signal processed
	225	611	Signal Number	Char(4)	The signal number that was processed.
	229	615	Handle action	Char(1)	The action taken on this signal.
					C Continue the process
					E Signal exception
					H Handle by invoking the signal catching function
					S Stop the process
					T Terminate the process
					U Terminate the request

Table 195. SG (Asychronous Signals) Journal Entries (continued). QASYSGJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	230	616	Signal Source	Char(1)	The source of the signal. **M** Machine source **P** Process source **Note:** When the signal source value is machine, the source job values are blank.
	231	617	Source Job Name	Char(10)	The first part of the source job's qualified name.
	241	627	Source Job User Name	Char(10)	The second part of the source job's qualified name.
	251	637	Source Job Number	Char(6)	The third part of the source jobs's qualified name.
	257	643	Source Job Current User	Char(10)	The current user profile for the source job.
	267	653	Generation Timestamp	Char(8)	The *DTS format of the time that the signal was generated. **Note:** The QWCCVTDT API can be used to convert a *DTS time stamp to other formats.

Table 196. SK (Secure Sockets Connections) Journal Entries. QASYSKJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	1	1			Heading fields common to all entry types. See Table 141 on page 501 and Table 142 on page 503 for field listing.
	224	610	Entry type	Char(1)	**A** Accept **C** Connect **D** DHCP address assigned **F** Filtered mail **P** Port unavailable **R** Reject mail **U** DHCP address denied
	225	611	Local IP Address	Char(15)	The local IP address.
	240	626	Local port	Char(5)	The local port.
	245	631	Remote IP Address	Char(15)	The remote IP address.
	260	646	Remote port	Char(5)	The remote port.
	265	651	Socket Descriptor	Bin(5)	The socket descriptor.
	269	655	Filter Description	Char(10)	The mail filter specified.
	279	665	Filter Data Length	Bin(4)	The length of the filter data.
	281	667	Filter Data[1]	Char(514)	The filter data.

Table 196. SK (Secure Sockets Connections) Journal Entries (continued). QASYSKJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	795	1181	Address Family	Char(10)	The address family. ***IPV4** Internet Protocol Version 4 ***IPV6** Internet Protocol Version 6
	805	1191	Local IP address	Char(46)	The local IP address.
	851	1237	Remote IP address [2]	Char(46)	The remote IP address
	897	1283	MAC address	Char(32)	The MAC address of the requesting client.
	929	1315	Host name	Char(255)	The host name of the requesting client.

[1] This is a variable length field. The first two bytes contain the length of the field.

[2] When the entry type is D, this field contains the IP address the DHCP server assigned the requesting client.

Table 197. SM (System Management Change) Journal Entries. QASYSMJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	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 **T** Access path recovery times changed
157	225	611	Access Type	Char(1)	**A** Add **C** Change **D** Delete **R** Remove **S** Display **T** Retrieve or receive
158	226	612	Sequence Number	Char(4)	Sequence number of the action
162	230	616	Message ID	Char(7)	Message ID associated with the action
169	237	623	Relational Database Name	Char(18)	Name of the relational database

Table 197. SM (System Management Change) Journal Entries (continued). QASYSMJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
187	255	641	File System Name	Char(10)	Name of the file system
197	265	651	Backup Option Changed	Char(10)	The backup option that was changed
207	275	661	Backup List Change	Char(10)	The name of the backup list that was changed
217	285	671	Network File Name	Char(10)	The name of the network file that was used
227	295	681	Network File Member	Char(10)	The name of the member of the network file
237	305	691	Network File Number	Zoned(6,0)	The number of the network file
243	311	697	Network File Owner	Char(10)	The name of the user profile that owns the network file
253	321	707	Network File Originating User	Char(8)	The name of the user profile that originated the network file
261	329	715	Network File Originating Address	Char(8)	The address that originated the network file

Table 198. SO (Server Security User Information Actions) Journal Entries. QASYSOJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry
					A Add entry
					C Change entry
					R Remove entry
157	225	611	User Profile	Char(10)	The name of the user profile.
	235	621	Server Authentication Entry	Char(1)	Y = Entry is a server authentication entry.
	236	622	Password Stored	Char(1)	**N** Password not stored
					S No change
					Y Password is stored.
	237	623	Server Name	Char(200)	The name of the server.
	437	823	(Reserved Area)	Char(3)	
	440	826	User ID Length	Binary (4)	The length of the user ID.
	442	828	(Reserved Area)	Char(20)	
	462	848	User ID	Char(1002)[1]	The ID for the user.

[1] This is a variable length field. The first 2 bytes contain the length of the field.

Table 199. ST (Service Tools Action) Journal Entries. QASYSTJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry
					A Service record
157	225	611	Service Tool	Char(2)	The type of entry.
					CS STRCPYSCN
					CD QTACTLDV
					CE QWTCTLTR
					CT DMPCLUTRC
					DC DLTCMNTRC
					DD DMPDLO
					DO DMPOBJ
					DS DMPSYSOBJ, QTADMPTS
					EC ENDCMNTRC
					ER ENDRMTSPT
					HD QYHCHCOP (DASD)
					HL QYHCHCOP (LPAR)
					PC PRTCMNTRC
					PE PRTERRLOG
					PI PRTINTDTA
					SE QWTSETTR
					SC STRCMNTRC
					SJ STRSRVJOB
					SR STRRMTSPT
					ST STRSST
					TA TRCTCPAPP
					TC TRCCNN (*FORMAT specified)
					TE ENDTRC, ENDPEX
					TI TRCINT or TRCCNN (*ON, *OFF, or *END specified)
					TS STRTRC, STRPEX
159	227	613	Object Name	Char(10)	Name of the object accessed
169	237	623	Library Name	Char(10)	Name of the library for the object
179	247	633	Object Type	Char(8)	Type of object
187	255	641	Job Name	Char(10)	The first part of the qualified job name
197	265	651	Job User Name	Char(10)	The second part of the qualified job name
207	275	661	Job Number	Zoned(6,0)	The third part of the qualified job name
213	281	667	Object Name	Char(30)	Name of the object for DMPSYSOBJ

Table 199. ST (Service Tools Action) Journal Entries (continued). QASYSTJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
243	311	697	Library Name	Char(30)	Name of the library for the object for DMPSYSOBJ
273	341	727	Object Type	Char(8)	Type of the object
281	349	735	DLO Name	Char(12)	Name of the document library object
293	361	747	(Reserved Area)	Char(8)	
301	369	755	Folder Path	Char(63)	The folder containing the document library object
	432	818	JUID Field	Char(10)	The JUID of the target job.
	442	828	Early Trace Action[1]	Char(10)	The action requested for early job tracing ***ON** Early tracing turned on ***OFF** Early tracing turned off ***RESET** Early tracing turned off and trace information deleted.
	452	838	Application Trace Option[2]	Char(1)	The trace option specified on TRCTCPAPP. **Y** Collection of trace information started **N** Collection of trace information stopped and trace information written to spooled file **E** Collection of trace information ended and all trace information purged (no output created)
	453	839	Application Traced[2]	Char(10)	The name of the application being traced.
	463	849	Service Tools Profile[3]	Char(10)	The name of the service tools profile used for STRSST.
		859	Source node ID	Char(8)	Source node ID
		867	Source user	Char(10)	Source user
		877	ASP name for object library	Char(10)	ASP name for object library
		887	ASP number for object library	Char(5)	ASP number for object library
		892	ASP name for DMPSYSOBJ object library	Char(10)	ASP name for DMPSYSOBJ object library
		902	ASP number for DMPSYSOBJ object library	Char(5)	ASP number for DMPSYSOBJ object library

[1] This field is used only when the entry type (offset 225) is CE.

[2] This field is used only when the entry type (offset 225) is TA.

[3] This field is used only when the entry type (offset 225) is ST.

Table 200. SV (Action to System Value) Journal Entries. QASYSVJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **A** Change to system values **B** Change to service attributes **C** Change to system clock
157	225	611	System Value or Service Attribute	Char(10)	The name of the system value or service attribute
167	235	621	New Value	Char(250)	The value to which the system value or service attribute was changed
417	485	871	Old Value	Char(250)	The value of the system value or service attribute before it was changed
667	735	1121	New Value Continued	Char(250)	Continuation of the value to which the system value or service attribute was changed.
917	985	1371	Old Value Continued	Char(250)	Continuation of the value of the system value or service attribute was changed.

Table 201. VA (Change of Access Control List) Journal Entries. QASYVAJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Status	Char(1)	Status of request. **S** Successful **F** Failed
157	225	611	Server Name	Char(10)	The name of the network server description that registered the event.
167	235	621	Server Date	Char(6)	The date the event was logged on the network server.
173	241	627	Server Time	Zoned(6,0)	The time the event was logged on the network server.
179	247	633	Computer Name	Char(8)	The name of the computer issuing the request to change the access control list.
187	255	641	Requester Name	Char(10)	The name of the user issuing the request.
197	265	651	Action Performed	Char(1)	The action performed on the access control profile: **A** Addition **C** Modification **D** Deletion
198	266	652	Resource Name	Char(260)	The name of the resource to be changed.

Table 202. VC (Connection Start and End) Journal Entries. QASYVCJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Connect Action.	Char(1)	The connection action that occurred. **S** Start **E** End **R** Reject
157	225	611	Server Name	Char(10)	The name of the network server description that registered the event.
167	235	621	Server Date	Char(6)	The date the event was logged on the network server.
173	241	627	Server Time	Zoned(6,0)	The time the event was logged on the network server.
179	247	633	Computer Name	Char(8)	The name of the computer associated with the connection request.
187	255	641	Connection User	Char(10)	The name of the user associated with the connection request.
197	265	651	Connect ID	Char(5)	The start or stop connection ID.
202	270	656	Rejection Reason	Char(1)	The reason the connection was rejected: **A** Automatic disconnect (timeout), share removed, or administrative permissions lacking **E** Error, session disconnect, or incorrect password **N** Normal disconnection or user name limit **P** No access permission to shared resource
203	271	657	Network Name	Char(12)	The network name associated with the connection.

Table 203. VF (Close of Server Files) Journal Entries. QASYVFJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Close Reason	Char(1)	The reason the file was closed. **A** Administrative disconnection **N** Normal client disconnection **S** Session disconnection
157	225	611	Server Name	Char(10)	The name of the network server description that registered the event.
167	235	621	Server Date	Char(6)	The date the event was logged on the network server.
173	241	627	Server Time	Zoned(6,0)	The time the event was logged on the network server.

Table 203. VF (Close of Server Files) Journal Entries (continued). QASYVFJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
179	247	633	Computer Name	Char(8)	The name of the computer requesting the close.
187	255	641	Connection User	Char(10)	The name of the user requesting the close.
197	265	651	File ID	Char(5)	The ID of the file being closed.
202	270	656	Duration	Char(6)	The number of seconds the file was open.
208	276	662	Resource Name	Char(260)	The name of the resource owning the accessed file.

Table 204. VL (Account Limit Exceeded) Journal Entries. QASYVLJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Reason	Char(1)	The reason the limit was exceeded.
					A Account expired
					D Account disabled
					L Logon hours exceeded
					U Unknown or unavailable
					W Workstation not valid
157	225	611	Server Name	Char(10)	The name of the network server description that registered the event.
167	235	621	Server Date	Char(6)	The date the event was logged on the network server.
173	241	627	Server Time	Zoned(6,0)	The time the event was logged on the network server.
179	247	633	Computer Name	Char(8)	The name of the computer with the account limit violation.
187	255	641	User	Char(10)	The name of the user with the account limit violation.
197	265	651	Resource Name	Char(260)	The name of the resource being used.

Table 205. VN (Network Log On and Off) Journal Entries. QASYVNJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Log Type	Char(1)	The type of event that occurred:
					F Logoff requested
					O Logon requested
					R Logon rejected

Table 205. VN (Network Log On and Off) Journal Entries (continued). QASYVNJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
157	225	611	Server Name	Char(10)	The name of the network server description that registered the event.
167	235	621	Server Date	Char(6)	The date the event was logged on the network server.
173	241	627	Server Time	Zoned(6,0)	The time the event was logged on the network server.
179	247	633	Computer Name	Char(8)	The name of the computer for the event.
187	255	641	User	Char(10)	The user who logged on or off.
197	265	651	User Privilege	Char(1)	Privilege of user logging on:
					A Administrator
					G Guest
					U User
198	266	652	Reject Reason	Char(1)	The reason the log on attempt was rejected:
					A Access denied
					F Forced off due to logon limit
					P Incorrect password
199	267	653	Additional Reason	Char(1)	Details of why access was denied:
					A Account expired
					D Account disabled
					L Logon hours not valid
					R Requester ID not valid
					U Unknown or unavailable

Table 206. VO (Validation List) Journal Entries. QASYVOJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	1	1			Heading fields common to all entry types. See Table 141 on page 501 andTable 142 on page 503 for field listing.
	224	610	Entry Type	Char(1)	The type of entry.
					A Add validation list entry
					C Change validation list entry
					F Find validation list entry
					R Remove validation list entry
					U Unsuccessful verify of a validation list entry
					V Successful verify of a validation list entry

Table 206. VO (Validation List) Journal Entries (continued). QASYVOJ4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	225	611	Unsuccessful Type	Char(1)	Type of unsuccessful verify. **E** Encrypted data is incorrect **I** Entry ID was not found **V** Validation list was not found
	226	612	Validation List	Char(10)	The name of the validation list.
	236	622	Library Name	Char(10)	The name of the library the validation list is in.
	246	632	Encrypted Data	Char(1)	Data value to be encrypted. **Y** Data to be encrypted was specified on the request. **N** Data to be encrypted was not specified on the request
	247	633	Entry Data	Char(1)	Entry data value. **Y** Entry data was specified on the request. **N** Entry data was not specified on the request.
	248	634	Entry ID Length	Binary(4)	The length of the entry ID.
	250	636	Data length	Binary(4)	The length of the entry data.
	252	638	Encrypted Data Attribute	Char (1)	Encrypted data. **' '** An encrypted data attribute was not specified. **0** The data to be encrypted can only be used to verify an entry. This is the default. **1** The data to be encrypted can be used to verify an entry and the data can be returned on a find operation.
	253	639	X.509 Certificate attribute	Char (1)	X.509 Certificate.
	254	640	(Reserved Area)	Char (28)	
	282	668	Entry ID	Byte(100)	The entry ID.
	382	768	Entry Data	Byte(1000)	The entry data.
		1768	ASP name for validation list library	Char(10)	ASP name for validation list library
		1778	ASP number for validation list library	Char(5)	ASP number for validation list library

Table 207. VP (Network Password Error) Journal Entries. QASYVPJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Error Type	Char(1)	The type of error that occurred. **P** Password error
157	225	611	Server Name	Char(10)	The name of the network server description that registered the event.
167	235	621	Server Date	Char(6)	The date the event was logged on the network server.
173	241	627	Server Time	Zoned(6,0)	The time the event was logged on the network server.
179	247	633	Computer Name	Char(8)	The name of the computer initiating the request.
187	255	641	User	Char(10)	The name of the user who attempted to log on.

Table 208. VR (Network Resource Access) Journal Entries. QASYVRJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Status	Char(1)	The status of the access. **F** Resource access failed **S** Resource access succeeded
157	225	611	Server Name	Char(10)	The name of the network server description that registered the event.
167	235	621	Server Date	Char(6)	The date the event was logged on the network server.
173	241	627	Server Time	Zoned(6,0)	The time the event was logged on the network server.
179	247	633	Computer Name	Char(8)	The name of the computer requesting the resource.
187	255	641	User	Char(10)	The name of the user requesting the resource.
197	265	651	Operation Type	Char(1)	The type of operation being performed: **A** Resource attributes modified **C** Instance of the resource created **D** Resource deleted **P** Resource permissions modified **R** Data read or run from a resource **W** Data written to resource **X** Resource was run
198	266	652	Return Code	Char(4)	The return code received if resource access is granted.
202	270	656	Server Message	Char(4)	The message code sent when access is granted.
206	274	660	File ID	Char(5)	The ID of the file being accessed.

Table 208. VR (Network Resource Access) Journal Entries (continued). QASYVRJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
211	279	665	Resource Name	Char(260)	Name of the resource being used.

Table 209. VS (Server Session) Journal Entries. QASYVSJE/J4/J5 field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Session Action	Char(1)	The session action that occurred.
					E End session
					S Start session
157	225	611	Server Name	Char(10)	The name of the network server description that registered the event.
167	235	621	Server Date	Char(6)	The date the event was logged on the network server.
173	241	627	Server Time	Zoned(6,0)	The time the event was logged on the network server.
179	247	633	Computer Name	Char(8)	The name of the computer requesting the session.
187	255	641	User	Char(10)	The name of the user requesting the session.
197	265	651	User Privilege	Char(1)	The privilege level of the user for session start:
					A Administrator
					G Guest
					U User
198	266	652	Reason Code	Char(1)	The reason code for ending the session.
					A Administrator disconnect
					D Automatic disconnect (timeout), share removed, or administrative permissions lacking
					E Error, session disconnect, or incorrect password
					N Normal disconnection or user name limit
					R Account restriction

Table 210. VU (Network Profile Change) Journal Entries. QASYVUJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.

Table 210. VU (Network Profile Change) Journal Entries (continued). QASYVUJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
156	224	610	Type	Char(1)	The type of record that was changed. **G** Group record **U** User record **M** User profile global information
157	225	611	Server Name	Char(10)	The name of the network server description that registered the event.
167	235	621	Server Date	Char(6)	The date the event was logged on the network server.
173	241	627	Server Time	Zoned(6,0)	The time the event was logged on the network server.
179	247	633	Computer Name	Char(8)	The name of the computer requesting the user profile change.
187	255	641	User	Char(10)	The name of the user requesting the user profile change.
197	265	651	Action	Char(1)	Action requested: **A** Addition **C** Change **D** Deletion **P** Incorrect password
198	266	652	Resource Name	Char(260)	Name of the resource.

Table 211. VV (Service Status Change) Journal Entries. QASYVVJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry: **C** Service status changed **E** Server stopped **P** Server paused **R** Server restarted **S** Server started
157	225	611	Server Name	Char(10)	The name of the network server description that registered the event.
167	235	621	Server Date	Char(6)	The date the event was logged on the network server.
173	241	627	Server Time	Zoned(6,0)	The time the event was logged on the network server.
179	247	633	Computer Name	Char(8)	The name of the computer requesting the change.
187	255	641	User	Char(10)	The name of the user requesting the change.

Table 211. VV (Service Status Change) Journal Entries (continued). QASYVVJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
197	265	651	Status	Char(1)	Status of the service request: **A** Service active **B** Start service pending **C** Continue paused service **E** Stop pending for service **H** Service pausing **I** Service paused **S** Service stopped
198	266	652	Service Code	Char(8)	The code of the service requested.
206	274	660	Text Set	Char(80)	The text being set by the service request.
286	354	740	Return Value	Char(4)	The return value from the change operation.
290	358	744	Service	Char(20)	The service that was changed.

Table 212. X0 (Network Authentication) Journal Entries. QASYX0JE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.

Table 212. X0 (Network Authentication) Journal Entries (continued). QASYX0JE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
156	224	610	Entry Type	Char(1)	The type of entry:
					1 Service ticket valid
					2 Service principals do not match
					3 Client principals do not match
					4 Ticket IP address mismatch
					5 Decryption of the ticket failed
					6 Decryption of authenticator failed
					7 Realm is not within client local realms
					8 Ticket is a replay attempt
					9 Ticket not yet valid
					A Decrypt of KRB_AP_PRIV or KRB_AP_SAFE checksum error
					B Remote IP address mismatch
					C Local IP address mismatch
					D KRB_AP_PRIV or KRB_AP_SAFE timestamp error
					E KRB_AP_PRIV or KRB_AP_SAFE replay error
					F KRB_AP_PRIV or KRB_AP_SAFE sequence order error
					K GSS accept — expired credential
					L GSS accept — checksum error
					M GSS accept — channel bindingst
					N GSS unwrap or GSS verify expired context
					O GSS unwrap or GSS verify decrypt/decode
					P GSS unwrap or GSS verify checksum error
					Q GSS unwrap or GSS verify sequence error
	225	611	Status Code	Char(8)	The status of the request
	233	619	GSS Status Value	Char(8)	GSS status value
	241	627	Remote IP Address	Char(21)	Remote IP address
	262	648	Local IP Address	Char(21)	Local IP address
	283	669	Encrypted Addresses	Char(256)	Encrypted IP addresses

Table 212. X0 (Network Authentication) Journal Entries (continued). QASYX0JE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	539	925	Encrypted Addresses Indicator	Char(1)	Encrypted IP addresses indicator **Y** all addresses included **N** not all addresses included **X** not provided
	540	926	Ticket flags	Char(8)	Ticket flags
	548	934	Ticket Authentication Time	Char(8)	Ticket authentication time
	556	942	Ticket Start Time	Char(8)	Ticket start time
	564	950	Ticket End Time	Char(8)	Ticket end time
	572	958	Ticket Renew Time	Char(8)	Ticket renew until time
	580	966	Message Time Stamp	Char(8)	X0E time stamp
	588	974	GSS Expiration Time Stamp	Char(8)	GSS credential expiration time stamp or context expiration time stamp
	596	982	Server Principal CCSID	Binary(5)	Server principal (from ticket) CCSID
	600	986	Server Principal Length	Binary(4)	Server principal (from ticket) length
	602	988	Server Principal Indicator	Char(1)	Server principal (from ticket) indicator **Y** server principal complete **N** server principal not complete **X** not provided
	603	989	Server Principal	Char(512)	Server principal (from ticket)
	1115	1501	Server Principal Parameter CCSID	Binary(5)	Server principal (from ticket) parameter CCSID
	1119	1505	Server Principal Parameter Length	Binary(4)	Server principal (from ticket) parameter length
	1121	1507	Server Principal Parameter Indicator	Char(1)	Server principal (from ticket) parameter indicator **Y** server principal complete **N** server principal not complete **X** not provided
	1122	1508	Server Principal Parameter	Char(512)	Server principal parameter that ticket must match
	1634	2020	Client Principal CCSID	Binary(5)	Client principal (from authenticator) CCSID
	1638	2024	Client Principal Length	Binary(4)	Client principal (from authenticator) length
	1640	2026	Client Principal Indicator	Char(1)	Client principal (from authenticator) indicator **Y** client principal complete **N** client principal not complete **X** not provided
	1641	2027	Client Principal	Char(512)	Client principal from authenticator

Table 212. X0 (Network Authentication) Journal Entries (continued). QASYX0JE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	2153	2539	Client Principal CCSID	Binary(5)	Client principal (from ticket) CCSID
	2157	2543	Client Principal Length	Binary(4)	Client principal (from ticket) length
	2159	2545	Client Principal Indicator	Char(1)	Client principal (from ticket) indicator **Y** client principal complete **N** client principal not complete **X** not provided
	2160	2546	Client Principal	Char(512)	Client principal from ticket
	2672	3058	GSS Server Principal CCSID	Binary(5)	Server principal (from GSS credential) CCSID
	2676	3062	GSS Server Principal Length	Binary(4)	Server principal (from GSS credential) length
	2678	3064	GSS Server Principal Indicator	Char(1)	Server principal (from GSS credential) indicator **Y** server principal complete **N** server principal not complete **X** not provided
	2679	3065	GSS Server Principal	Char(512)	Server principal from GSS credential
	3191	3577	GSS Local Principal CCSID	Binary(5)	GSS local principal name CCSID
	3195	3581	GSS Local Principal Length	Binary(4)	GSS local principal name length
	3197	3583	GSS Local Principal Indicator	Char(1)	GSS local principal name indicator **Y** local principal complete **N** local principal not complete **X** not provided
	3198	3584	GSS Local Principal	Char(512)	GSS local principal
	3710	4096	GSS Remote Principal CCSID	Binary(5)	GSS remote principal name CCSID
	3714	4100	GSS Remote Principal Length	Binary(4)	GSS remote principal name length
	3716	4102	GSS Remote Principal Indicator	Char(1)	GSS remote principal name indicator **Y** remote principal complete **N** remote principal not complete **X** not provided
	3717	4103	GSS Remote Principal	Char(512)	GSS remote principal

Table 213. YC (Change to DLO Object) Journal Entries. QASYYCJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	Object access **C** Change of a DLO object
157	225	611	Object Name	Char(10)	Name of the object
167	235	621	Library Name	Char(10)	Name of the library
177	245	631	Object Type	Char(8)	Type of object
185	253	639	Office User	Char(10)	User profile of the office user
195	263	649	Folder or Document Name	Char(12)	Name of the document or folder
207	275	661	(Reserved Area)	Char(8)	
215	283	669	Folder Path	Char(63)	The folder containing the document library object
278	346	732	On Behalf of User	Char(10)	User working on behalf of another user
288	356	742	Access Type	Packed(5,0)	Type of access [1]

1 See Table 218 on page 597 for a list of the codes for access types.

Table 214. YR (Read of DLO Object) Journal Entries. QASYYRJE/J4/J5 Field Description File

Offstes					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	Object access **R** Read of a DLO object
157	225	611	Object Name	Char(10)	Name of the object
167	235	621	Library Name	Char(10)	Name of the library
177	245	631	Object Type	Char(8)	Type of object
185	253	639	Office User	Char(10)	User profile of the office user
195	263	649	Folder or Document Name	Char(12)	Name of the document library object
207	275	661	(Reserved Area)	Char(8)	
215	283	669	Folder Path	Char(63)	The folder containing the document library object
278	346	732	On Behalf of User	Char(10)	User working on behalf of another user
288	356	742	Access Type	Packed(5,0)	Type of access [1]

1 See Table 218 on page 597 for a list of the codes for access types.

Table 215. ZC (Change to Object) Journal Entries. QASYZCJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	Object access **C** Change of an object
157	225	611	Object Name	Char(10)	Name of the object
167	235	621	Library Name	Char(10)	Name of the library in which the object is located
177	245	631	Object Type	Char(8)	Type of object
185	253	639	Access Type	Packed(5,0)	Type of access [1]
188	256	642	Access Specific Data	Char(50)	Specific data about the access
238			(Reserved Area)	Char(20)	
	306	692	(Reserved Area)	Char(18)	
	324	710	Object Name Length [2]	Binary (4)	The length of the object name.
258	326	712	Object Name CCSID[2]	Binary(5)	The coded character set identifier for the object name.
262	330	716	Object Name Country or Region ID[2]	Char(2)	The Country or Region ID for the object name.
264	332	718	Object Name Language ID[2]	Char(3)	The language ID for the object name.
267	335	721	(Reserved area)	Char(3)	
270	338	724	Parent File ID[2, 3]	Char(16)	The file ID of the parent directory.
286	354	740	Object File ID[2, 3]	Char(16)	The file ID of the object.
302	370	756	Object Name[2]	Char(512)	The name of the object.
	882	1268	Object File ID	Char(16)	The file ID of the object.
	898	1284	ASP Name[6]	Char(10)	The name of the ASP device.
	908	1294	ASP Number[6]	Char(5)	The number of the ASP device.
	913	1299	Path Name CCSID	Binary(5)	The coded character set identifier for the absolute path name.
	917	1303	Path Name Country or Region ID	Char(2)	The Country or Region ID for the absolute path name
	919	1305	Path Name Language ID	Char(3)	The language ID for the absolute path name.
	922	1308	Path Name Length	Binary(4)	The length of the absolute path name.
	924	1310	Path Name Indicator	Char(1)	The absolute path name indicator: **Y** The Absolute Path Name field contains an absolute path name for the object. **N** The Absolute Path Name field does not contain an absolute path name for the object.

Table 215. ZC (Change to Object) Journal Entries (continued). QASYZCJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	925	1311	Relative File ID[4]	Char(16)	The relative file ID of the absolute path name.
	941	1327	Absolute Path Name[5]	Char(5002)	The absolute path name of the object.

[1] See Table 218 on page 597 for a list of the codes for access types.

[2] These fields are used only for objects in the QOpenSys, "root" file systems, and user-defined file systems.

[3] An ID that has the left-most bit set and the rest of the bits zero indicates that the ID is NOT set.

[4] When the path name indicator (offset 924) is ″N″, this field will contain the relative file ID of the absolute path name. When the path name indicator is ″Y″, this field will contain 16 bytes of hex zeroes.

[5] This is a variable length field. The first 2 bytes contain the length of the path name.

[6] If the object is in a library, this is the ASP information of the object's library. If the object is not in a library, this is the ASP information of the object.

Table 216. ZM (SOM Method Access) Journal Entries. QASYZMJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1				Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224		Access Type	Char(1)	Type of access
157	225		Object Existence	Char(1)	Y Object existence
158	226		Object Management	Char(1)	Y Object management
159	227		Object Operational	Char(1)	Y Object operational
160	228		Object Alter	Char(1)	Y Object alter
161	229		Object Reference	Char(1)	Y Object reference
162	230		Reserved	Char(10)	Reserved field
172	240		List Management	Char(1)	Y Authorization list management
173	241		Read	Char(1)	Y Read
174	242		Add	Char(1)	Y Add
175	243		Update	Char(1)	Y Update
176	244		Delete	Char(1)	Y Delete
177	245		Execute	Char(1)	Y Execute
178	246		Reserved	Char(10)	Reserved field
188	256		Class File ID	Char(16)	File ID of class
204	272		Object FIle ID	Char(16)	File ID of object
220	288		Method Name	Char(4096)	Name of Method

Table 217. ZR (Read of Object) Journal Entries. QASYZRJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
1	1	1			Heading fields common to all entry types. See Table 141 on page 501,Table 142 on page 503, and Table 143 on page 504 for field listing.
156	224	610	Entry Type	Char(1)	Object access **R** Read of an object
157	225	611	Object Name	Char(10)	Name of the object
167	235	621	Library Name	Char(10)	Name of the library in which the object is located
177	245	631	Object Type	Char(8)	Type of object
185	253	639	Access Type	Packed(5,0)	Type of access [1]
188	256	642	Access Specific Data	Char(50)	Specific data about the access
238			(Reserved Area)	Char(20)	
	306	692	(Reserved Area)	Char(18)	
	324	710	Object Name Length [2]	Binary(4)	The length of the object name.
258	326	712	Object Name CCSID[2]	Binary(5)	The coded character set identifier for the object name.
262	330	716	Object Name Country or Region ID[2]	Char(2)	The Country or Region ID for the object name.
264	332	718	Object Name Language ID[2]	Char(3)	The language ID for the object name.
267	335	721	(Reserved area)	Char(3)	
270	338	724	Parent File ID[2,3]	Char(16)	The file ID of the parent directory.
286	354	740	Object File ID[2,3]	Char(16)	The file ID of the object.
302	370	756	Object Name[2]	Char(512)	The name of the object.
	882	1268	Object File ID	Char(16)	The file ID of the object.
	898	1284	ASP Name	Char(10)	The name of the ASP device.
	908	1294	ASP Number	Char(5)	The number of the ASP device.
	913	1299	Path Name CCSID	Binary(5)	The coded character set identifier for the absolute path name.
	917	1303	Path Name Country or Region ID	Char(2)	The Country or Region ID for the absolute path name
	919	1305	Path Name Language ID	Char(3)	The language ID for the absolute path name.
	922	1308	Path Name Length	Binary(4)	The length of the absolute path name.
	924	1310	Path Name Indicator	Char(1)	The absolute path name indicator: **Y** The Absolute Path Name field contains an absolute path name for the object. **N** The Absolute Path Name field does not contain an absolute path name for the object.
	925	1311	Relative File ID[4]	Char(16)	The relative file ID of the absolute path name.
	941	1327	Absolute Path Name[5]	Char(5002)	The absolute path name of the object.

Table 217. ZR (Read of Object) Journal Entries (continued). QASYZRJE/J4/J5 Field Description File

Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**

[1] See Table 218 for a list of the codes for access types.

[2] These fields are used only for objects in the QOpenSys, "root" file systems, and user-defined file systems.

[3] An ID that has the left-most bit set and the rest of the bits zero indicates that the ID is NOT set.

[4] When the path name indicator (offset 924) is ″N″, this field will contain the relative file ID of the absolute path name. When the path name indicator is ″Y″, this field will contain 16 bytes of hex zeroes.

[5] This is a variable length field. The first 2 bytes contain the length of the path name.

Table 218 lists the access codes used for object auditing journal entries in files QASYYCJE, QASYYRJE, QASYZCJE, and QASYZRJE.

Table 218. Numeric Codes for Access Types

Code	Access Type	Code	Access Type	Code	Access Type
1	Add	24	Hold	47	Save with Storage Free
2	Activate Program	25	Initialize	48	Save and Delete
3	Analyze	26	Load	49	Submit
4	Apply	27	List	50	Set
5	Call or TFRCTL	28	Move	51	Send
6	Configure	29	Merge	52	Start
7	Change	30	Open	53	Transfer
8	Check	31	Print	54	Trace
9	Close	32	Query	55	Verify
10	Clear	33	Reclaim	56	Vary
11	Compare	34	Receive	57	Work
12	Cancel	35	Read	58	Read/Change DLO Attribute
13	Copy	36	Reorganize	59	Read/Change DLO Security
14	Create	37	Release	60	Read/Change DLO Content
15	Convert	38	Remove	61	Read/Change DLO all parts
16	Debug	39	Rename	62	Add Constraint
17	Delete	40	Replace	63	Change Constraint
18	Dump	41	Resume	64	Remove Constraint
19	Display	42	Restore	65	Start Procedure
20	Edit	43	Retrieve	66	Get Access on **OOPOOL
21	End	44	Run	67	Sign object
22	File	45	Revoke	68	Remove all signatures
23	Grant	46	Save	69	Clear a signed object

Appendix G. Commands and Menus for Security Commands

This appendix describes the commands and menus for security tools. Examples of how to use the commands are included throughout this manual.

Two menus are available for security tools:

- The SECTOOLS (Security Tools) menu to run commands interactively.
- The SECBATCH (Submit or Schedule Security Reports to Batch) menu to run the report commands in batch. The SECBATCH menu has two parts. The first part of the menu uses the Submit Job (SBMJOB) command to submit reports for immediate processing in batch.

  The second part of the menu uses the Add Job Schedule Entry (ADDJOBSCDE) command. You use it to schedule security reports to be run regularly at a specified day and time.

Options on the Security Tools Menu

Following is the part of the SECTOOLS menu that relates to user profiles. To access this menu, type `GO SECTOOLS`

```
SECTOOLS                        Security Tools

Select one of the following:

  Work with profiles
     1. Analyze default passwords

     2. Display active profile list
     3. Change active profile list
     4. Analyze profile activity

     5. Display activation schedule
     6. Change activation schedule entry

     7. Display expiration schedule
     8. Change expiration schedule entry
```

Table 219 describes these menu options and the associated commands:

Table 219. Tool Commands for User Profiles

Menu[1] Option	Command Name	Description	Database File Used
1	ANZDFTPWD	Use the Analyze Default Passwords command to report on and take action on user profiles that have a password equal to the user profile name.	QASECPWD[2]
2	DSPACTPRFL	Use the Display Active Profile List command to display or print the list of user profiles that are exempt from ANZPRFACT processing.	QASECIDL[2]

Table 219. Tool Commands for User Profiles (continued)

Menu[1] Option	Command Name	Description	Database File Used
3	CHGACTPRFL	Use the Change Active Profile List command to add and remove user profiles from the exemption list for the ANZPRFACT command. A user profile that is on the active profile list is permanently active (until you remove the profile from the list). The ANZPRFACT command does not disable a profile that is on the active profile list, no matter how long the profile has been inactive.	QASECIDL[2]
4	ANZPRFACT	Use the Analyze Profile Activity command to disable user profiles that have not been used for a specified number of days. After you use the ANZPRFACT command to specify the number of days, the system runs the ANZPRFACT job nightly. You can use the CHGACTPRFL command to exempt user profiles from being disabled.	QASECIDL[2]
5	DSPACTSCD	Use the Display Profile Activation Schedule command to display or print information about the schedule for enabling and disabling specific user profiles. You create the schedule with the CHGACTSCDE command.	QASECACT[2]
6	CHGACTSCDE	Use the Change Activation Schedule Entry command to make a user profile available for sign on only at certain times of the day or week. For each user profile that you schedule, the system creates job schedule entries for the enable and disable times.	QASECACT[2]
7	DSPEXPSCDE	Use the Display Expiration Schedule command to display or print the list of user profiles that are scheduled to be disabled or removed from the system in the future. You use the CHGEXPSCDE command to set up user profiles to expire.	QASECEXP[2]
8	CHGEXPSCDE	Use the Change Expiration Schedule Entry command to schedule a user profile for removal. You can remove it temporarily (by disabling it) or you can delete it from the system. This command uses a job schedule entry that runs every day at 00:01 (1 minute after midnight). The job looks at the QASECEXP file to determine whether any user profiles are set up to expire on that day. Use the DSPEXPSCD command to display the user profiles that are scheduled to expire.	QASECEXP[2]
9	PRTPRFINT	Use the Print Profile Internals command to print a report of internal information on the number of entries in a user profile (*USRPRF) object.	

Notes:

1. Options are from the SECTOOLS menu.
2. This file is in the QUSRSYS library.

You can page down on the menu to see additional options. Table 220 describes the menu options and associated commands for security auditing:

Table 220. Tool Commands for Security Auditing

Menu[1] Option	Command Name	Description	Database File Used
10	CHGSECAUD	Use the Change Security Auditing command to set up security auditing and to change the system values that control security auditing. When you run the CHGSECAUD command, the system creates the security audit (QAUDJRN) journal if it does not exist. The CHGSECAUD command provides options that make it simpler to set the QAUDLVL (audit level) system value. You can specify *ALL to activate all of the possible audit level settings. Or, you can specify *DFTSET to activate the most commonly used settings (*AUTFAIL, *CREATE, *DELETE, *SECURITY, and *SAVRST). **Note:** If you use the security tools to set up auditing, be sure to plan for management of your audit journal receivers. Otherwise, you might quickly encounter problems with disk utilization.	
11	DSPSECAUD	Use the Display Security Auditing command to display information about the security audit journal and the system values that control security auditing.	

Notes:

1. Options are from the SECTOOLS menu.

How to Use the Security Batch Menu

Following is the first part of the SECBATCH menu:

```
SECBATCH         Submit or Schedule Security Reports To Batch
                                                                    System:
Select one of the following:

  Submit Reports to Batch
     1. Adopting objects
     2. Audit journal entries
     3. Authorization list authorities
     4. Command authority
     5. Command private authorities
     6. Communications security
     7. Directory authority
     8. Directory private authority
     9. Document authority
    10. Document private authority
    11. File authority
    12. File private authority
    13. Folder authority
```

When you select an option from this menu, you see the Submit Job (SBMJOB) display, such as the following:

```
                              Submit Job (SBMJOB)
Type choices, press Enter.

Command to run . . . . . > PRTADPOBJ USRPRF(*ALL
_____________________________________________
_____________________________________________
_____________________________________________...
Job name . . . . . .   *JOBD          Name, *JOBD
Job description  . . . . . .   *USRPRF       Name, *USRPRF
  Library  . . . . . .                        Name, *LIBL, *CURLIB
Job queue  . . . . . .   *JOBD          Name, *JOBD
  Library  . . . . . .                        Name, *LIBL, *CURLIB
Job priority (on JOBQ) . . . . .   *JOBD          1-9, *JOBD
Output priority (on OUTQ)  . . .   *JOBD          1-9, *JOBD
Print device . . . . . .   *CURRENT      Name, *CURRENT, *USRPRF...
```

If you want to change the default options for the command, you can press F4 (Prompt) on the *Command to run* line.

To see the Schedule Batch Reports, page down on the SECBATCH menu. By using the options on this part of the menu, you can, for example, set up your system to run changed versions of reports regularly.

```
SECBATCH         Submit or Schedule Security Reports To Batch
                                                              System:
Select one of the following:

    28. User objects
    29. User profile information
    30. User profile internals
    31. Check object integrity

  Schedule Batch Reports
    40. Adopting objects
    41. Audit journal entries
    42. Authorization list authorities
    43. Command authority
    44. Command private authority
    45. Communications security
    46. Directory authority
```

You can page down for additional menu options. When you select an option from this part of the menu, you see the Add Job Schedule Entry (ADDJOBSCDE) display:

```
                      Add Job Schedule Entry (ADDJOBSCDE)

Type choices, press Enter.

Job name . . . . . .   _____                  Name, *JOBD
Command to run . . . . . > PRTADPOBJ USRPRF(*ALL)
_____________________________________________
_____________________________________________
_____________________________________________
_____________________________________________...
Frequency  . . . . . .   _______*ONCE, *WEEKLY, *MONTHLY
Schedule date, or  . . . . . .   *CURRENT      Date, *CURRENT, *MONTHST
Schedule day . . . . . .   *NONE         *NONE, *ALL, *MON, *TUE.
+ for more values
Schedule time  . . . . . .   *CURRENT      Time, *CURRENT
```

You can position your cursor on the *Command to run* line and press F4 (Prompt) to choose different settings for the report. You should assign a meaningful job name so that you can recognize the entry when you display the job schedule entries.

Options on the Security Batch Menu

Table 221 describes the menu options and associated commands for security reports.

When you run security reports, the system prints only information that meets both the selection criteria that you specify and the selection criteria for the tool. For example, job descriptions that specify a user profile name are security-relevant. Therefore, the job description (PRTJOBDAUT) report prints job descriptions in the specified library only if the public authority for the job description is not *EXCLUDE *and* if the job description specifies a user profile name in the USER parameter.

Similarly, when you print subsystem information (PRTSBSDAUT command), the system prints information about a subsystem only when the subsystem description has a communications entry that specifies a user profile.

If a particular report prints less information than you expect, consult the online help information to find out the selection criteria for the report.

Table 221. Commands for Security Reports

Menu[1] Option	**Command Name**	**Description**	**Database File Used**
1, 40	PRTADPOBJ	Use the Print Adopting Objects command to print a list of objects that adopt the authority of the specified user profile. You can specify a single profile, a generic profile name (such as all profiles that begin with Q), or all user profiles on the system. This report has two versions. The full report lists all adopted objects that meet the selection criteria. The changed report lists differences between adopted objects that are currently on the system and adopted objects that were on the system the last time that you ran the report.	QSECADPOLD[2]
2, 41	DSPAUDJRNE	Use the Display Audit Journal Entries command to display or print information about entries in the security audit journal. You can select specific entry types, specific users, and a time period.	QASYxxJE[3]

Table 221. Commands for Security Reports (continued)

Menu[1] Option	Command Name	Description	Database File Used
3, 42	PRTPVTAUT *AUTL	When you use the Print Private Authorities command for *AUTL objects, you receive a list of all the authorization lists on the system. The report includes the users who are authorized to each list and what authority the users have to the list. Use this information to help you analyze sources of object authority on your system. This report has three versions. The full report lists all authorization lists on the system. The changed report lists additions and changes to authorization since you last ran the report. The deleted report lists users whose authority to the authorization list has been deleted since you last ran the report. When you print the full report, you have the option to print a list of objects that each authorization list secures. The system will create a separate report for each authorization list.	QSECATLOLD[2]
6, 45	PRTCMNSEC	Use the Print Communications Security command to print the security-relevant settings for objects that affect communications on your system. These settings affect how users and jobs can enter your system. This command produces two reports: a report that displays the settings for configuration lists on the system and a report that lists security-relevant parameters for line descriptions, controllers, and device descriptions. Each of these reports has a full version and a changed version.	QSECCMNOLD[2]
15, 54	PRTJOBDAUT	Use the Print Job Description Authority command to print a list of job descriptions that specify a user profile and have public authority that is not *EXCLUDE. The report shows the special authorities for the user profile that is specified in the job description. This report has two versions. The full report lists all job description objects that meet the selection criteria. The changed report lists differences between job description objects that are currently on the system and job description objects that were on the system the last time that you ran the report.	QSECJBDOLD[2]

Table 221. Commands for Security Reports (continued)

Menu[1] Option	Command Name	Description	Database File Used
See note 4	PRTPUBAUT	Use the Print Publicly Authorized Objects command to print a list of objects whose public authority is not *EXCLUDE. When you run the command, you specify the type of object and the library or libraries for the report. Use the PRTPUBAUT command to print information about objects that every user on the system can access. This report has two versions. The full report lists all objects that meet the selection criteria. The changed report lists differences between the specified objects that are currently on the system and objects (of the same type in the same library) that were on the system the last time that you ran the report.	QPBxxxxxx[5]
See note 4.	PRTPVTAUT	Use the Print Private Authorities command to print a list of the private authorities to objects of the specified type in the specified library. Use this report to help you determine the sources of authority to objects. This report has three versions. The full report lists all objects that meet the selection criteria. The changed report lists differences between the specified objects that are currently on the system and objects (of the same type in the same library) that were on the system the last time that you ran the report. The deleted report lists users whose authority to an object has been deleted since you last printed the report.	QPVxxxxxx[5]
24, 63	PRTQAUT	Use the Print Queue Report to print the security settings for output queues and job queues on your system. These settings control who can view and change entries in the output queue or job queue. This report has two versions. The full report lists all output queue and job queue objects that meet the selection criteria. The changed report lists differences between output queue and job queue objects that are currently on the system and output queue and job queue objects that were on the system the last time that you ran the report.	QSECQOLD[2]

Table 221. Commands for Security Reports (continued)

Menu[1] Option	Command Name	Description	Database File Used
25, 64	PRTSBSDAUT	Use the Print Subsystem Description command to print the security-relevant communications entries for subsystem descriptions on your system. These settings control how work can enter your system and how jobs run. The report prints a subsystem description only if it has communications entries that specify a user profile name. This report has two versions. The full report lists all subsystem description objects that meet the selection criteria. The changed report lists differences between subsystem description objects that are currently on the system and subsystem description objects that were on the system the last time that you ran the report.	QSECSBDOLD[2]
26, 65	PRTSYSSECA	Use the Print System Security Attributes command to print a list of security-relevant system values and network attributes. The report shows the current value and the recommended value.	
27, 66	PRTTRGPGM	Use the Print Trigger Programs command to print a list of trigger programs that are associated with database files on your system. This report has two versions. The full report lists every trigger program that is assigned and meets your selection criteria. The changed report lists trigger programs that have been assigned since the last time that you ran the report.	QSECTRGOLD[2]
28, 67	PRTUSROBJ	Use the Print User Objects command to print a list of the user objects (objects not supplied by IBM) that are in a library. You might use this report to print a list of user objects that are in a library (such as QSYS) that is in the system portion of the library list. This report has two versions. The full report lists all user objects that meet the selection criteria. The changed report lists differences between user objects that are currently on the system and user objects that were on the system the last time that you ran the report.	QSECPUOLD[2]
29, 68	PRTUSRPRF	Use the Print User Profile command to analyze user profiles that meet specified criteria. You can select user profiles based on special authorities, user class, or a mismatch between special authorities and user class. You can print authority information, environment information, or password information.	
30, 69	PRTPRFINT	Use the Print Profile Internals command to print a report of internal information on the number of entries contained in a user profile (*USRPRF) object.	

Table 221. Commands for Security Reports (continued)

Menu[1] Option	Command Name	Description	Database File Used
31, 70	CHKOBJITG	Use the Check Object Integrity command to determine whether operable objects (such as programs) have been changed without using a compiler. This command can help you to detect attempts to introduce a virus program on your system or to change a program to perform unauthorized instructions.	

Notes:

1. Options are from the SECBATCH menu.
2. This file is in the QUSRSYS library.
3. xx is the two-character journal entry type. For example, the model output file for AE journal entries is QSYS/QASYAEJE. The model output files are described in Appendix F of this book.
4. The SECTOOLS menu contains options for the object types that are typically of concern to security administrators. For example, use options 11 or 50 to run the PRTPUBAUT command against *FILE objects. Use the general options (18 and 57) to specify the object type. Use options 12 and 51 to run the PRTPVTAUT command against *FILE objects. Use the general options (19 and 58) to specify the object type.
5. The xxxxxx in the name of the file is the object type. For example, the file for program objects is called QPBPGM for public authorities and QPVPGM for private authorities. The files are in the QUSRSYS library.

   The file contains a member for each library for which you have printed the report. The member name is the same as the library name.

Commands for Customizing Security

Table 222 describes the commands that you can use to customize the security on your system. These commands are on the SECTOOLS menu:

Table 222. Commands for Customizing Your System

Menu[1] Option	Command Name	Description	Database File Used
60	CFGSYSSEC	Use the Configure System Security command to set security-relevant system values to their recommended settings. The command also sets up security auditing on your system. "Values That Are Set by the Configure System Security Command" describes what the command does.	
61	RVKPUBAUT	Use the Revoke Public Authority command to set the public authority to *EXCLUDE for a set of security-sensitive commands on your system. "What the Revoke Public Authority Command Does" on page 609 lists the actions that the RVKPUBAUT command performs.	

Notes:

1. Options are from the SECTOOLS menu.

Values That Are Set by the Configure System Security Command

Table 223 on page 608 lists the system values that are set when you run the CFGSYSSEC command. The CFGSYSSEC command runs a program that is called QSYS/QSECCFGS.

Table 223. Values Set by the CFGSYSSEC Command

System Value Name	Setting	System Value Description
QAUTOCFG	0 (No)	Automatic configuration of new devices
QAUTOVRT	0	The number of virtual device descriptions that the system will automatically create if no device is available for use.
QALWOBJRST	*NONE	Whether system state programs and programs that adopt authority can be restored
QDEVRCYACN	*DSCMSG (Disconnect with message)	System action when communications is re-established
QDSCJOBITV	120	Time period before the system takes action on a disconnected job
QDSPSGNINF	1 (Yes)	Whether users see the sign-on information display
QINACTITV	60	Time period before the system takes action on an inactive interactive job
QINACTMSGQ	*ENDJOB	Action that the system takes for an inactive job
QLMTDEVSSN	1 (Yes)	Whether users are limited to signing on at one device at a time
QLMTSECOFR	1 (Yes)	Whether *ALLOBJ and *SERVICE users are limited to specific devices
QMAXSIGN	3	How many consecutive, unsuccessful sign-on attempts are allowed
QMAXSGNACN	3 (Both)	Whether the system disables the workstation or the user profile when the QMAXSIGN limit is reached.
QRMTSIGN	*FRCSIGNON	How the system handles a remote (pass-through or TELNET) sign-on attempt.
QRMTSVRATR	0 (Off)	Allows the system to be analyzed remotely.
QSECURITY [1]	50	The level of security that is enforced
QPWDEXPITV	60	How often users must change their passwords
QPWDMINLEN	6	Minimum length for passwords
QPWDMAXLEN	8	Maximum length for passwords
QPWDPOSDIF	1 (Yes)	Whether every position in a new password must differ from the same position in the last password
QPWDLMTCHR	See note 2	Characters that are not allowed in passwords
QPWDLMTAJC	1 (Yes)	Whether adjacent numbers are prohibited in passwords
QPWDLMTREP	2 (Cannot be repeated consecutively)	Whether repeating characters in are prohibited in passwords
QPWDRQDDGT	1 (Yes)	Whether passwords must have at least one number
QPWDRQDDIF	1 (32 unique passwords)	How many unique passwords are required before a password can be repeated
QPWDVLDPGM	*NONE	The user exit program that the system calls to validate passwords

Notes:

1. If you are currently running with a QSECURITY value of 30 or lower, be sure to review the information in Chapter 2 of this book before you change to a higher security level.
2. The restricted characters are stored in message ID CPXB302 in the message file QSYS/QCPFMSG. They are shipped as AEIOU@$#. You can use the Change Message Description (CHGMSGD) command to change the restricted characters.

The CFGSYSSEC command also sets the password to *NONE for the following IBM-supplied user profiles:

QSYSOPR
QPGMR
QUSER
QSRV
QSRVBAS

Finally, the CFGSYSSEC command sets up security auditing according to the values that you have specified by using the Change Security Auditing (CHGSECAUD) command.

Changing the Program

If some of these settings are not appropriate for your installation, you can create your own version of the program that processes the command. Do the following:

__ Step 1. Use the Retrieve CL Source (RTVCLSRC) command to copy the source for the program that runs when you use the CFGSYSSEC command. The program to retrieve is QSYS/QSECCFGS. When you retrieve it, give it a *different name*.

__ Step 2. Edit the program to make your changes. Then compile it. When you compile it, make sure that you *do not* replace the IBM-supplied QSYS/QSECCFGS program. Your program should have a different name.

__ Step 3. Use the Change Command (CHGCMD) command to change the program to process command (PGM) parameter for the CFGSYSSEC command. Set the PGM value to the name of your program. For example, if you create a program in the QGPL library that is called MYSECCFG, you would type the following:

```
CHGCMD CMD(QSYS/CFGSYSSEC) PGM(QGPL/MYSECCFG)
```

Note: If you change the QSYS/QSECCFGS program, IBM cannot guarantee or imply reliability, serviceability, performance or function of the program. The implied warranties of merchantability and fitness for a particular purpose are expressly disclaimed.

What the Revoke Public Authority Command Does

You can use the Revoke Public Authority (RVKPUBAUT) command to set the public authority to *EXCLUDE for a set of commands and programs. The RVKPUBAUT command runs a program that is called QSYS/QSECRVKP. As it is shipped, the QSECRVKP revokes public authority (by setting public authority to *EXCLUDE) for the commands that are listed in Table 224 on page 610 and the application programming interfaces (APIs) that are listed in Table 225 on page 610. When your system arrives, these commands and APIs have their public authority set to *USE.

The commands that are listed in Table 224 on page 610 and the APIs that are listed in Table 225 on page 610 all perform functions on your system that may provide an opportunity for mischief. As security administrator, you should explicitly authorize users to run these commands and programs rather than make them available to all system users.

When you run the RVKPUBAUT command, you specify the library that contains the commands. The default is the QSYS library. If you have more than one national

language on your system, you need to run the command for each QSYSxxx library.

Table 224. Commands Whose Public Authority Is Set by the RVKPUBAUT Command

ADDAJE	CHGJOBQE	RMVCMNE
ADDCFGLE	CHGPJE	RMVJOBQE
ADDCMNE	CHGRTGE	RMVPJE
ADDJOBQE	CHGSBSD	RMVRTGE
ADDPJE	CHGWSE	RMVWSE
ADDRTGE	CPYCFGL	RSTLIB
ADDWSE	CRTCFGL	RSTOBJ
CHGAJE	CRTCTLAPPC	RSTS36F
CHGCFGL	CRTDEVAPPC	RSTS36FLR
CHGCFGLE	CRTSBSD	RSTS36LIBM
CHGCMNE	ENDRMTSPT	STRRMTSPT
CHGCTLAPPC	RMVAJE	STRSBS
CHGDEVAPPC	RMVCFGLE	WRKCFGL

The APIs in Table 225 are all in the QSYS library:

Table 225. Programs Whose Public Authority Is Set by the RVKPUBAUT Command

QTIENDSUP
QTISTRSUP
QWTCTLTR
QWTSETTR
QY2FTML

On V3R7, when you run the RVKPUBAUT command, the system sets the public authority for the root directory to *USE (unless it is already *USE or less).

Changing the Program

If some of these settings are not appropriate for your installation, you can create your own version of the program that processes the command. Do the following:

__ Step 1. Use the Retrieve CL Source (RTVCLSRC) command to copy the source for the program that runs when you use the RVKPUBAUT command. The program to retrieve is QSYS/QSECRVKP. When you retrieve it, give it a *different name*.

__ Step 2. Edit the program to make your changes. Then compile it. When you compile it, make sure that you *do not* replace the IBM-supplied QSYS/QSECRVKP program. Your program should have a different name.

__ Step 3. Use the Change Command (CHGCMD) command to change the program to process command (PGM) parameter for the RVKPUBAUT command. Set the PGM value to the name of your program. For example, if you create a program in the QGPL library that is called MYRVKPGM, you would type the following:

```
CHGCMD CMD(QSYS/RVKPUBAUT) PGM(QGPL/MYRVKPGM)
```

Note: If you change the QSYS/QSECRVKP program, IBM cannot guarantee or imply reliability, serviceability, performance or function of the program. The implied warranties of merchantability and fitness for a particular purpose are expressly disclaimed.

Appendix H. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. 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 IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described 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:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or region or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country or region where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created

programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator
3605 Highway 52 N
Rochester, MN 55901-7829
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information 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.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

Advanced Function Printing
APPN
Application System/400
AS/400
AS/400e
C/400
CallPath/400
DB2 for OS/400
DRDA
e (Stylized)
FFST
IBM
Integrated Language Environment
iSeriesLotus
Lotus Domino
LPDA
Operating System/400
Operational Assistant
OS/2
OS/400
PrintManager
Print Service Facility
RPG/400
SecureWay
SQL/400
SystemView
System/36
System/38
400

C-bus is a trademark of Corollary, Inc.

Microsoft, Windows, Windows NT, and the Windows 95 logo are registered trademarks of Microsoft Corporation.

Java and HotJava are trademarks of Sun Microsystems, Inc.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

Other company, product, and service names may be trademarks or service marks of others.

Related information

You may need to refer to other IBM books for more specific information about a particular topic. The following IBM iSeries books contain information that you may need.

Advanced Security

- *Tips and Tools for Securing Your iSeries*, SC41-5300-06, provides a set of practical suggestions for using the security features of iSeries and for establishing operating procedures that are security–conscious. This book also describes how to set up and use security and use security tools that are part of OS/400. See the iSeries: Information Center Supplemental Manuals CD-ROM.
- *Implementing iSeries 400 Security, 3rd Edition* by Wayne Madden and Carol Woodbury. Loveland, Colorado: 29th Street Press, a division of Duke Communication International, 1998. Provides guidance and practical suggestions for planning, setting up, and managing your iSeries security.

  ISBN Order Number
  1–882419–78–2

Backup and Recovery

- *Backup and Recovery*, SC41-5304-06, provides information about planning a backup and recovery strategy, saving information from your system, and recovering your system, how to use journaling, commitment control, auxiliary storage pools, and disk protection options. See the iSeries: Information Center Supplemental Manuals CD-ROM.
- Additional backup and recovery information can be found in the Information Center. See "Prerequisite and related information" on page xvi for more information.

Basic Security Information and Physical Security

- The Basic System Security and Planning topic in the Information Center explains why security is necessary, defines major concepts, and provides information on planning, implementing, and monitoring basic security on the system. See "Prerequisite and related information" on page xvi for details.

iSeries Access for Windows Licensed Program

- The iSeries Access for Windows topic in the Information Center provides technical information about the iSeries Access for Windows programs for all versions of iSeries Access for Windows. See "Prerequisite and related information" on page xvi for details.

Communications and Networking

- *SNA Distribution Services*, SC41-5410-01, 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.
- *Remote Work Station Support*, SC41-5402-00, 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. See the iSeries: Information Center Supplemental Manuals CD-ROM.
- The Information Center 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. See "Prerequisite and related information" on page xvi for details.
- The Information Center provides information that describes how to use and configure TCP/IP and the several TCP/IP applications, such as FTP, SMTP, and TELNET. See "Prerequisite and related information" on page xvi for details.

Cryptography

- *Cryptographic Support/400*, SC41-3342-00, describes the data security capabilities of the Cryptographic Facility licensed program product. It explains how to use the facility and provides reference information for programmers. See the iSeries: Information Center Supplemental Manuals CD-ROM.

General System Operations

- ″Basic system operations″ in the Information Center provides information about how to start and stop the system and work with system problems. See “Prerequisite and related information” on page xvi for more details.

IBM-Supplied Program Installation and System Configuration

- *Local Device Configuration*, SC41-5121-00, provides information about how to do an initial configuration and how to change that configuration. It also contains conceptual information about device configuration. See the iSeries: Information Center Supplemental Manuals CD-ROM.
- *Software Installation*, SC41-5120-06, provides step-by-step procedures for initial install, installing licensed programs, program temporary fixes (PTFs), and secondary languages from IBM. See the iSeries: Information Center Supplemental Manuals CD-ROM.

Integrated File System

- The File Systems and Management topic in the Information Center provides an overview of the integrated file system, including what it is, how it might be used, and what interfaces are available. See “Prerequisite and related information” on page xvi for details.

The Internet

- *AS/400 Internet Security: Protecting Your AS/400 from HARM on the Internet* SG24–4929 discusses the security issues and the risk associated with connecting your iSeries to the Internet. It provides examples, recommendations, tips, and techniques for applications.
- *iSeries and the Internet*, G325–6321, helps you address potential security concerns you may have when connecting your iSeries to the Internet. For more information, visit the following IBM I/T (Information Technology) Security home page:

  `http://www.ibm.com/security`
- *Cool Title About the AS/400 and Internet*, SG24–4815, can help you understand and then use the Internet (or your own intranet) from your iSeries. It helps you to understand how to use the functions and features. This book helps you to get started quickly using e-mail, file transfer, terminal emulation, gopher, HTTP, and 5250 to HTML Gateway.

IBM Lotus Domino

- The URL, http://notes.net/notesua.nsf, provides information on Lotus Notes, Domino, and IBM Domino for iSeries. From this web site, you can download information in Domino database (.NSF) and Adobe Acrobat (.PDF) format, search databases, and find out how to obtain printed manuals.

Migration and System/36 Environment

- *System/38 Migration Planning*, SC41-4153-00, provides information to help migrate products and applications using the 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 iSeries Migration Aid.

Optical Support

- *Optical Support*, SC41-5310-03, provides information on functions that are unique for *Optical Support*. It also contains helpful information for the use and understanding of; CD-Devices, Directly attached Optical Media Library Devices, and LAN attached Optical Media Library Devices. See the iSeries: Information Center Supplemental Manuals CD-ROM.

Printing

- The Information Center provides information on printing elements and concepts of the system, printer file and print spooling support for printing operation, and printer connectivity. See "Prerequisite and related information" on page xvi for details.

Programming

- *CL Programming*, SC41-5721-05, provides a wide-ranging discussion of 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. See the iSeries: Information Center Supplemental Manuals CD-ROM.
- The CL topic in the Information Center (see "Prerequisite and related information" on page xvi for details) provides a description of all the iSeries 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 licensed programs, including all the various languages and utilities—are described in other books that support those licensed programs.
- The Programming topic in the Information Center provides information about many of the languages and utilities available on the iSeries. It contains summaries of:
  - All iSeries 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).
- The Information Center contains several topics regarding System Management and Work Management on the iSeries. Some of these topics include performance data collection, system values management, and storage management. For details on accessing the Information Center, see "Prerequisite and related information" on page xvi.
- *Work Management*, SC41-5306-03, provides information about how to create and change a work management environment. See the iSeries: Information Center Supplemental Manuals CD-ROM.
- The API topic in the Information Center (see "Prerequisite and related information" on page xvi for details) provides information on how to create, use, and delete objects that help manage system performance, use spooling efficiently, and maintain database files efficiently. This book 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.

Utilities

- *WebSphere Development Studio: Application Development Manager User's Guide*, SC09-2133-02, 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 book 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.
- *ADTS for AS/400: Source Entry Utility*, SC09-2605-00, provides information about using the Application Development Tools source entry utility (SEU) to create and edit source members. The book explains how to start and end an SEU session and how to use the many features of this full-screen text editor. The book 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.See the iSeries: Information Center Supplemental Manuals CD-ROM.

- The DB2 Universal Database for iSeries topic in the Information Center provides an overview of how to design, write, run, and test SQL/400* statements. It also describes interactive Structured Query Language (SQL), and provides examples of how to write SQL statements in COBOL, RPG, C, FORTRAN, and PL/I programs. See "Prerequisite and related information" on page xvi for details.
- The DB2 Universal Database for iSeries topic in the Information Center 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

  See "Prerequisite and related information" on page xvi for details.

Index

Special Characters

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

Readers' Comments — We'd Like to Hear from You

iSeries
Security Reference
Version 5

Publication No. SC41-5302-06

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

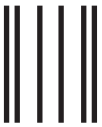
Company or Organization

Phone No.

Readers' Comments — We'd Like to Hear from You
SC41-5302-06

Cut or Fold
Along Line

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

IBM CORPORATION
ATTN DEPT 542 IDCLERK
3605 HWY 52 N
ROCHESTER MN 55901-7829

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

Printed in U.S.A.

SC41-5302-06