IBM i
7.4

Security
Security reference

SC41-5302-14

Note

Before using this information and the product it supports, read the information in “Notices” on page 913.

This edition applies to IBM® i 7.4 (product number 5770-SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CISC models.

This edition replaces SC41-5302-14.

This document may contain references to Licensed Internal Code. Licensed Internal Code is Machine Code and is licensed to you under the terms of the IBM License Agreement for Machine Code.

Contents

What's new for IBM i 7.4

Read about new or significantly changed information for the Security reference topic collection.

Authority Collection has been enhanced to support authority collection for objects. The enhanced support includes the following changes.

Updated commands:

- Start Authority Collection (STRAUTCOL)
- End Authority Collection (ENDAUTCOL)
- Delete Authority Collection (DLTAUTCOL)

New command:

- Change Authority Collection (CHGAUTCOL)

New SQL views:

- QSYS2.AUTHORITY_COLLECTION_OBJECT
- QSYS2.AUTHORITY_COLLECTION_LIBRARIES
- QSYS2.AUTHORITY_COLLECTION_FSOBJ
- QSYS2.AUTHORITY_COLLECTION_DLO

Information has been added for Mitigating Spectre and Meltdown vulnerabilities in new and existing programs.

Miscellaneous updates to audit journal entries have been made.

Other miscellaneous updates have been made to this topic collection.

How to see what's new or changed

To help you see where technical changes have been made, the information center uses:

- The » image to mark where new or changed information begins.
- The « image to mark where new or changed information ends.

In PDF files, you might see revision bars (| or +) in the left margin of new and changed information.

Chapter 1. Introduction to IBM i security

The IBM Systems family covers a wide range of users. Security on the IBM i platform is flexible enough to meet the requirements of this wide range of users and situations.

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. You need to understand the features and options available so that you can adapt them to your own security requirements.

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 topic collection 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.

Related information

Planning physical security

Security level

The IBM i platform offers five levels of security. You can choose which level of security you want the system to enforce by setting the security level (QSECURITY) system value.

Level 10:
Level 10 is no longer supported.

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 IBM i platforms with high security requirements, and it is designed to meet Common Criteria (CC) security requirements.

Related reference

Using System Security (QSecurity) system value
You can choose how much security you want the system to enforce by setting the security level (QSECURITY) system value.

System values

System values provide customization on many characteristics of your IBM i platform. You can use system values to define system-wide security settings.

For example, you can specify the following settings:

- 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.

Related concepts

Security 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.

Signing

You can reinforce integrity by signing software objects that you use.

A key component of security is *integrity*: being able to trust that objects on the system have not been tampered with or altered. Your IBM i operating system software is protected by digital signatures.

Signing your software object 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 system are signed).

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.

Related information

Using digital signatures to protect software integrity
Digital Certificate Manager

Single sign-on enablement

Single *sign-on* is an authentication process in which a user can access more than one system by entering a single user ID and password. In today's heterogeneous networks with partitioned systems and multiple platforms, administrators must cope with the complexities of managing identification and authentication for network users.

To enable a single sign-on environment, IBM provides two technologies that work together to enable users to sign in with their Windows user name and password and be authenticated to IBM i platforms in the network. Network Authentication Service (NAS) and Enterprise Identity Mapping (EIM) are the two technologies that an administrator must configure to enable a single sign-on environment. Windows operating systems, AIX®, and z/OS® use Kerberos protocol to authenticate users to the network. A secure, centralized system, called a key distribution center, authenticates principals (Kerberos users) to the network.

While Network Authentication Service (NAS) allows a IBM i platform to participate in the 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 IBM i user name, can also be associated with this EIM identifier. When a user signs on to the network and accesses a IBM i platform, that user is not prompted for a user ID and password. If the Kerberos authentication is successful, applications can look up the association to the EIM identifier to find the IBM i user name. The user no longer needs a password to sign on to IBM i platform 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 (NAS) and Enterprise Identity Mapping (EIM) on your system.

Related information

Scenario: Creating a single signon test environment

User profiles

On the IBM i operating system, 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. The following list describes some of the 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.

Related concepts
User profiles
User profiles are a powerful and flexible tool. Designing them well can help you protect your system and customize it for your users.

Group profiles

A *group profile* is a special type of user profile. Rather than giving authority to each user individually, you can use a group profile to define authority for a group of users.

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.

Related concepts
Planning group profiles
A group profile is a useful tool when several users have similar security requirements. You can directly create group files or you can make an existing profile into a group profile. When you use group profiles, you can manage authority more efficiently and reduce the number of individual private authorities for objects.

Group ownership of objects
This topic provides detailed information about the group ownership of objects.

Primary group for an object
You can specify a primary group for an object.

Copying user profiles
You can create a user profile by copying another user profile or a group profile.

Resource security

The ability to access an object is called *authority*. Resource security on the IBM i operating system enables you to control object authorities by defining who can use which objects and how those objects can be used.

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. The following list describes 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 in 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. Correct 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. You can use SQL statements to manage this authority.

Related concepts

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

Security audit journal

You can use security audit journals to audit the effectiveness of security on your system.

The IBM i operating 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.

Related concepts

Auditing security on IBM i
This section describes techniques for auditing the effectiveness of security on your system.

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* (iASP) 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. Functional changes to independent disk pools have security implications on your system. For example, when you perform a CRTUSRPRF, you cannot 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 support many library-based objects and user-defined file systems. However, several objects are not allowed on independent disk pools.

Related information

Supported and unsupported object types

Chapter 2. Using System Security (QSecurity) system value

You can choose how much security you want the system to enforce by setting the security level (QSECURITY) system value.

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

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

Levels of security

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 12.

If you want to change the security level, use the Work with System Values (WRKSYSVAL) command. The minimum security level you should use is 40. Security level 40 and 50 provide the system integrity protection required to run a secure server. Security levels 30 and below do not provide the integrity protection required for a secure operating environment. The change takes effect the next time you perform an initial program load (IPL). Table 1 on page 7 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

Table 1. Security levels: function comparison (continued)				
Function	**Level 20**	**Level 30**	**Level 40**	**Level 50**
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
Direct access to all objects using object address.	Yes	No	No	No
User profile created automatically.	No	No	No	No
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 is enforced for all storage.	No	No	Yes	Yes
Library QTEMP is a temporary object.	No	No	No	No
*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.

Default special authorities

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.

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 Security system values.

Table 2 on page 9 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 83 and “Special authority” on page 89 provide more information about user classes and special authorities.

Considerations

At security level 30 the system does not automatically give users access to all resources. At lower security levels, all users are given *ALLOBJ special authority.

Security level 30 is not a secure level to run your production system. At security level 30 and below, users can directly call system level interfaces that are not intended to be directly called by user applications. In addition, user applications can access internal control blocks and object contents directly using an address. This is a security and integrity exposure. At security level 30, the integrity protection mechanisms are not activated to the same level as security level 40 and 50. Therefore, security level 40 or higher is strongly recommended.

Security level 40 and 50 provide significant integrity protection that is not available on security level 30 and below. To run a secure server, you must run at security level 40 or 50. Security level 40 and 50 are similar in capabilities. This was not always the case but, over time, the capabilities that were initially available in security level 50 have been moved into the security level 40 support. There are still some differences between 40 and 50. The differences are mostly internal processing of buffers and control blocks plus the restrictions on how messages can be sent within a job, see "Restricting message handling" on page 20. Running security level 50 provides the most secure level to run your server.

Related concepts

Security level
The IBM i platform offers five levels of security. You can choose which level of security you want the system to enforce by setting the security level (QSECURITY) system value.

Related tasks

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

Security level 10

At security level 10, you have no security protection. Therefore, security level 10 is not recommended. Running your server at this security level is both a security and integrity risk as you do not have the protection of the higher security levels, 40 and 50, activated and being enforced.

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, "IBM-supplied user profiles," on page 345 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 almost 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

Security level 20 provides more security functions than level 10. However, because at security level 20 all profiles are created with *ALLOBJ special authority by default, security level 20 is not recommended either. Running your server at this security level is both a security and integrity risk as you do not have the protection of the higher security levels, 40 and 50, activated and being enforced.

Security 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.

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.

Consider performing the following list of recommended 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 "Default values for user profiles" on page 345 in Appendix B, "IBM-supplied user profiles," on page 345 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.

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.

Refer to Table 2 on page 9 to see how special authorities differ between level 20 and higher security levels.

Security level 30

Security level 30 provides more security features than security level 20. Security level 30 is not considered a secure level as the integrity protection features available on security level 40 and 50 are not activated at security level 30. Running your server at this security level is both a security and integrity risk as you do not have the protection of the higher security levels, 40 and 50, activated and being enforced.

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 to update special authorities the next time you perform an initial program load (IPL).

Special authorities that the user was given at 10 or 20, but didn't 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 9 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. Consider performing the following recommended activities:

- For each application, set the appropriate authorities for application objects.
- Test each application by using either actual user profiles or special test user profiles.
  - Remove *ALLOBJ special authority from the user profiles that are 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 the test profiles, grant appropriate authorities for application objects to the production user profiles that should have access to the application.
- 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. You can 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.

Related reference

Defining how information can be accessed
You can define what operations can be preformed on objects, data, and fields.

Security level 40

Security level 40 prevents potential integrity or security risks from programs that can circumvent security in special cases. Security level 50 provides enhanced integrity protection for installations with strict security requirements.

Table 3 on page 12 compares how security functions are supported at levels 30, 40, and 50.

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]; operation fails.	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 the system area of disk that is defined as read-only or no access.	Attempt may succeed.	AF journal entry; [1] operation fails.	AF journal entry; [1] operation fails.
An attempt is made to restore a program that does not have a validation value. [2]	No validation is performed. Program must be converted before it can be used.	No validation is performed. Program must be converted before it can be used.	No validation is performed. Program must be converted 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 change a program's associated space.	Attempt is successful.	AF journal entry;[1] operation fails.	AF journal entry;[1] operation fails.

Table 3. Comparison of security levels 30, 40, and 50 (continued)

Scenario description	Level 30	Level 40	Level 50
An attempt is made to change a job's address space.	Attempt is successful.	AF journal entry;[1] operation fails.	AF journal entry;[1] operation fails.
A user state program attempts to call or transfer control to a system domain program.	AF journal entry [1]	AF journal entry;[1] operation fails.	AF journal entry;[1] operation fails.
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 call 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.
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, 3] operation fails.[3]	AF journal entry;[1, 3] operation fails. [3]

1 An authority failure (AF) type entry is written to the audit (QAUDJRN) journal, if the auditing function is active. See Chapter 9, "Auditing security on IBM i," on page 259 for more information about the audit function.

2 Programs created before Version 1 Release 3 do not have a validation value.

3 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 12, 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 or higher, the system prevents attempts to directly call system programs that are 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. You can access objects that are *SYSTEM domain by 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 by using the Display Program (DSPPGM) command. You can display the state of a service program by using the Display Service Program (DSPSRVPGM) command.

Table 4 on page 14 shows the domain and state access rules:

Table 4. Domain and state access

Program state	Object domain	
	***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.

The supported interfaces for object access are by CL commands and IBM i APIs. For example, accessing a DB2® file using the Open API or "read/add/update/delete record" API (which are also built into most runtime support) are examples of supported interfaces to access a *FILE object. Using SQL, DDM, DRDA, or any other IBM i provided interface is also a supported interface. Using CL commands such as CHGPF is also an example of a supported interface. The same is true for other object types, examples of supported interfaces for *PGM objects include calling a program, displaying a program using DSPPGM, and compiling a program. If the applications are developed with the supported interfaces and compiler statements and directives, your application, at ALL security levels, will run without system integrity errors. Security level 40 and 50 issues will be encountered if unsupported interfaces are used or direct object access is attempted by the application

"Direct" access to an object or IBM i control block, using the object address and then reading or modifying the bytes within the object, is NOT allowed. Using programming languages that support pointer/address access is typically how one could attempt the direct access. The C programming language is typically how one would attempt this type of access but pointer/address access is also available in other languages. Resolving (or the application runtime code finding the object within a library) and then setting a pointer to the internals of the object is NOT allowed on security level 40 and 50 but is allowed on security level 30 for most object types. Also, if a user program were to use an already "set" pointer that addresses a protected control block or object, and tries to read or write from this control block or object directly, then an exception will occur on security level 40 and 50 and the read or write would be denied. For most objects or control blocks, this access will work on security level 30. Resolved or set pointers/addresses to certain objects and control blocks exist within a users job. This is how one may obtain an already set pointer to a protected control block or object. An audit can occur on level 30 as "object domain" is applied to all objects and thus an audit record will be sent to the audit journal for violations that would occur on security level 40 and 50. See "Enhanced hardware storage protection" on page 16 for a description of control blocks that aren't "objects" so there is no object domain checking and no auditing available. The Object Domain setting for an object is displayed via DSPOBJD, *FULL display.

Within the Machine Interface, high-level instructions exist. Many of these instructions are available to user level programs and are generated by the compilers when programs are created. But, as a part of the MI instruction set, many of these instructions are reserved for operating system use (need extra privilege to use them). These restricted instructions are never generated by the underlying compile process of user code. They are only allowed when running an IBM i operating system program. HOWEVER, it is possible to

patch or alter a program, by using the service tools or offline, to “add” one of these restricted instructions to a program’s instruction stream. This is why there is support in the system to “block” the use of these instructions from a user level program (these instructions, if used incorrectly, could cause serious issues in the system). There is also a CL command, CHKOBJITG (Check Object Integrity), that will look for these patched programs on the system and report them. The restore process will also look for patched programs during restore and will either remove the patch, audit the restore, or allow the patched program on the system (admin options using QFRCCVNRST, QVFYOBJRST and QALWOBJRST syst89iopem values).

Journal entry:

When the following conditions are met, an authority failure (AF) entry, violation type D or R, is written to the QAUDJRN journal:

- The auditing function is active
- The QAUDLVL system value includes *PGMFAIL.
- 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 run under that user profile. Thus an unauthorized user might submit a job to run under the user profile specified in the job description.

At security level 40 and higher, the job fails unless the user submitting the job has *USE authority to both the job description and the user profile that is specified in the job description. At security level 30, the job runs if the submitter has *USE authority to the job description. The submitter does not need to have *USE authority to the user profile specified in the job description.

This is the issue that happens most frequently when moving from level 30 to 40 or 50. Within the job description object, a user can be named on the USER parameter. When on security level 30, when this particular job description is used during a submit job, the authority check is simply “does the user submitting the job have *USE authority to the job description”. On security level 40 and 50, the same authority check is done but an extra check is made to see whether the user submitting the job has *USE authority to the user profile specified in the job description. This extra check, which can be easily fixed, causes most of the issues when moving to a higher security level. By default, when a user profile is created, the *PUBLIC authority is set to *EXCLUDE. This prevents a user from submitting a job to run under a different user profile by specifying to use the user profile in the job description. To solve this problem, the security administrator can grant authority for any user who should be allowed to submit the job to run under the user profile that is specified in the job description.

To allow USER1 to submit a job that runs under user JOBDUSER do the following:

- GRTOBJAUT OBJ(JOBDUSER) OBJTYPE(*USRPRF) USER(USER1) AUT(*USE)
- SBMJOB CMD(CALL PGM(TEST)) JOB(TEST) USER(*JOBD)

To find all *JOBD objects that contain a user profile name, signon as a security officer and run the PRTJOBDAUT LIB(*ALL) command.

Journal entry:

When the following conditions are met, an AF entry, violation type J, is written to the QAUDJRN journal:

- The auditing function is active
- The QAUDLVL system value includes *AUTFAIL
- A user submits a job, while the user is not authorized to the user profile in the job description

Signing on without a user ID and password

Your security level determines how the system controls 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.

Journal entry:

When the following conditions are met, an AF entry, violation type S, is written to the QAUDJRN journal:

- The auditing function is active
- The QAUDLVL system value includes *AUTFAIL
- A user attempts to sign on without entering a user ID and password and the subsystem description allows it

Note that the attempt fails at security level 40 and higher.

Related concepts

Subsystem descriptions
The subsystem descriptions perform several functions on the system.

Enhanced hardware storage protection

Enhanced hardware storage protection allows blocks of system information that are located on the memory 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.

Enhanced hardware storage protection is supported on all IBM i models.

All IBM i objects, *FILE, *PGM, *JOBD, *CMD, etc. have an object domain. Object domain protection is a capability that is detected in software thus it allows a domain violation audit to occur on level 30. On level 30 the system can detect the domain setting and send an audit record when a user state program tries to access a system domain object. Enhanced Hardware Storage Protection (HSP) is different. HSP is detected by the Power® hardware and cannot be detected when the protection is turned off for an object or control block. This powerful protection is either on or off for an object or control block. On security level 30, for most objects and control blocks, it is off. It is on for everything on security level 40 and 50. There is no way to audit HSP violations on security level 30 thus the need to test your applications on security level 40 or 50. The good thing about HSP for IBM i objects and control blocks that are used by the operating system is that the objects also have a domain (so you get the security level 30 domain violation audit records). However, there are many lower-level control blocks, which are used by the Licensed Internal Code, that are not IBM i objects thus do not have an object domain (but are protected, at 40 and 50, by HSP). If an application was patched to access one of these control blocks, it fails at 40 and 50 but works at 30 (without an audit). The good thing about HSP is that when you get to security level 40 or 50, you have industry leading protection for your objects. But, to get there, you need to test on security level 40 and 50 and cannot rely on audit on security level 30 to find every potential issue.

Journal entry:

When the following conditions are met, an AF entry, violation type R, is written to the QAUDJRN journal:

- The auditing function is active
- The QAUDLVL system value includes *PGMFAIL
- A program attempts to write to an area of memory protected by the enhanced hardware storage protection feature

Protecting a program's associated space

For original program model (OPM) programs, at security level 40 and higher, the associated space of a program object cannot be directly changed by user state programs. For integrated language environment

(ILE) programs, the associated space of a program object cannot be changed by user state programs at any security level.

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 IBM i operating system are system state programs in the user domain. 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 that is 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 Common Criteria level of security. You might notice some performance effect because of this additional checking.

Parameter validation is checking done by every IBM i API. APIs are defined as user domain, system state which makes them directly callable by user applications. These are the interfaces that are called directly by user applications. Parameter Validation is checking done by the IBM i API program to test the parameters that are passed by the user application to the system state IBM i program. Each parameter is tested to ensure that both the parameter value itself (typically a pointer to the actual parameter string) as well as the value are in storage that is read/write to the user application. If the parameter and value are in storage that the user application has access to, then everything is fine. If the parameter or value is in storage that the user application cannot access, the parameter validation signals an error message and will not continue. This checking is on for security level 40 and 50 only and not on for security level 30 (and no auditing is done on security level 30). This checking is necessary to prevent a user application from tricking a system program into writing over storage that the user application would not have access to. This could be done by passing a parameter, to a "return value", that addresses protected storage and have the system program write over the control block when setting the "return value" (because the system program has access as it runs with higher privilege than the user program). Without parameter validation, the system control blocks would be at risk of being compromised and thus the system would not function correctly if the control block contained "bad" data.

Validation of programs being restored

When a program is created, the 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 system takes action according to the Force Conversion on Restore (QFRCCVNRST) and Allow Object Restore (QALWOBJRST) system values.

In addition to a validation value, a program might optionally have a digital signature that can be verified on 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), QFRCCVNRST and 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.

Note: System state programs must have a valid IBM digital signature. Otherwise, they cannot be restored, no matter how the system values are set

The first filter is the 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, the QFRCCVNRST system value. With this system value you specify whether 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, the QALWOBJRST system value. This system value controls whether objects with security sensitive attributes can be restored.

Notes:

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

Related reference

Security-related system values
This topic introduces the security-related system values on your IBM i operating system.

Changing to security level 40

Before migrating to level 40, make sure that all of your applications run successfully at security level 30. Security level 30 gives you the opportunity to test resource security for all of your applications.

Follow these steps 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 299 gives complete instructions for setting up the auditing function.
2. Make sure that 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 of your applications at security level 30. Pay particular attention to the following detailed entries in AF type entries:

   B
   Restricted (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 before 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

You might want to move back to level 30 from level 40 temporarily because you need to test new applications for integrity errors. Or, you might 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 9.) This removes resource security protection.

Security level 50

Security level 50 is designed to meet some of the requirements defined by the Controlled Access Protection Profile (CAPP) for Common Criteria (CC) compliance. Security level 50 provides enhanced integrity protection, in addition to what is provided by security level 40, for installations with strict security requirements.

The security functions included for security level 50 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

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 write 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.
- 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 26 for more information.

Related tasks

Changing to security level 50

If your current security level is 10 or 20, change the security level to 40 before you change it to 50. If your current security level is 30 or 40, you need to evaluate the QALWUSRDMN value and recompile some programs to prepare for security level 50.

Restricting message handling

Messages sent between programs provide the potential for integrity exposures.

At security level 50, you are able to restrict the messages sent between programs to protect the integrity of your system.

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 need to be the program called by the system state program. For example, in this call 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, 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

If your current security level is 10 or 20, change the security level to 40 before you change it to 50. If your current security level is 30 or 40, you need to evaluate the QALWUSRDMN value and recompile some programs to prepare for 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 before 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 the QALWUSRDMN system value. Controlling user domain objects is important to system integrity.
- 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.

Related concepts

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.

Disabling security level 50

After changing to security level 50, you might find you need to move back to security level 30 or 40 temporarily. For example, you might need to test new applications for integrity errors; or you might 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.

Related reference

Using System Security (QSecurity) system value
You can choose how much security you want the system to enforce by setting the security level (QSECURITY) system value.

Chapter 3. Security 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.

You can restrict users from changing the security-related system values. System service tools (SST) and dedicated service tools (DST) provide an option to lock these system values. By locking the system values, you 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 certificates to digital certificate store with the Add Verifier API and restrict password resetting on the digital certificate store.

Note: If you lock the security-related system values and need to perform a restore operation as part of a system recovery, be aware that you need to unlock the system values to complete the restore operation. This ensures that the system values are free to be changed during the initial program load (IPL).

You can restrict the following system values by using the lock option:

- QALWJOBITP
- QALWOBJRST
- QALWUSRDMN
- QAUDCTL
- QAUDENDACN
- QAUDFRCLVL
- QAUDLVL
- QAUDLVL2
- QAUTOCFG
- QAUTORMT
- QAUTOVRT
- QCRTAUT
- QCRTOBJAUD
- QDEVRCYACN
- QDSPSGNINF
- QDSCJOBITV
- QFRCCVNRST
- QINACTMSGQ
- QLMTDEVSSN
- QLMTSECOFR
- QMAXSGNACN
- QMAXSIGN
- QPWDCHGBLK
- QPWDEXPITV
- QPWDEXPWRN
- QPWDLMTAJC
- QPWDLMTCHR
- QPWDLMTREP
- QPWDLVL
- QPWDMAXLEN

- QPWDMINLEN
- QPWDPOSDIF
- QPWDRQDDGT
- QPWDRQDDIF
- QPWDRULES
- QPWDVLDPGM
- QRETSVRSEC
- QRMTSIGN
- QRMTSRVATR
- QSCANFS
- QSCANFSCTL
- QSECURITY
- QSHRMEMCTL
- QUSEADPAUT
- QVFYOBJRST

You can use system service tools (SST) or dedicated service tools (DST) to lock and unlock the security-related system values. However, you must use DST if you are in recovery mode because SST is not available during this mode. Otherwise, use SST to lock or unlock the security-related system values.

To lock or unlock security-related system values with the Start System Service Tools (STRSST) command, follow these steps:

Note: You must have a service tools user ID and password to lock or unlock the security-related system values.

1. Open a character-based interface.
2. On the command line, type STRSST.
3. Type your service tools user ID and password.
4. Select option 7 (Work with system security).
5. Type 1 to unlock security-related system values or 2 to lock security-related system values in the **Allow system value security changes** parameter.

To lock or unlock security-related system values using dedicated service tools (DST) during an attended IPL of a system recovery, follow these steps:

1. From the IPL or Install the System display, select option 3 (Use Dedicated Service Tools).

   Note: This step assumes that you are in recovery mode and are performing an attended IPL.
2. Sign on to DST using your service tools user ID and password.
3. Select option 13 (Work with system security).
4. Type 1 to unlock security-related system values or 2 to lock security-related system values in the **Allow system value security changes** parameter.

Related concepts

System values

System values provide customization on many characteristics of your IBM i platform. You can use system values to define system-wide security settings.

General security system values

This topic introduces the general system values that you can use to control security on your IBM i operating system.

Overview:

General security system values allow you to set security function to support the decisions you made when developing your security policy. For example, in your security policy you state that systems containing confidential information, such as customer accounts or payroll inventories, need a stricter level of security than systems used for testing applications that are developed within your company. You can then plan and set a security level on these systems that corresponds with the decisions you made while developing your security policy.

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

Note:
Changes take effect immediately. IPL is required only when changing the security level (QSECURITY system value) or password level (QPWDLVL system value).

General system values that control security on your system are as follows:

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

QSCANFS
Scan file systems

QSCANFSCTL
Scan file systems control

QSECURITY
Security level

QSHRMEMCTL
Shared memory control

QUSEADPAUT
Use Adopted Authority

QVFYOBJRST
Verify object on restore

Allow User Domain Objects (QALWUSRDMN)

All objects are assigned a domain attribute when they are created. A domain is a characteristic of an object that controls how programs can access the object. The Allow User Domain Objects (QALWUSRDMN) system value specifies which libraries are allowed to contain user domain objects of type *USRSPC, *USRIDX, and *USRQ.

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. The restriction does not apply to user domain objects of type program (*PGM), server program (*SRVPGM), and SQL packages (*SQLPKG).

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 5. Possible values for the QALWUSRDMN system value:

Value	Description
*ALL	User domain objects are allowed in all libraries and directories on the system. This is the shipped value.
[*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.

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 can use MI interface to read or change the data in user domain objects in these libraries without being audited.

Note: If you run the Reclaim Storage (RCLSTG) command, user domain objects might need to be moved in and out of the QRCL (reclaim storage) library. To run the RCLSTG command successfully, you might 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 Authority for New Objects (QCRTAUT) system value specifies the public authority for a newly created object.

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.

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 6. Possible values for the QCRTAUT system value:	
Value	**Description**
*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 might encounter problems with signing on at new or automatically created devices. To avoid these problems when you change QCRTAUT to something other than *CHANGE, make sure 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 Display Sign-On Information (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 password verifications that were not valid
- The number of days until the password expires (if the password is due to expire within the password expiration warning days (QPWDEXPWRN)))

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

Password verifications not valid . . . . . :   3

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

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 7. Possible values for the QDSPSGNINF system value:	
Value	**Description**
0	Display is not shown.
1	Display is shown.

Recommended value: 1 (Display is shown) is recommended so that 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 Inactive Job Time-Out Interval (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 in display wait (DSPW) status, 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 IBM i 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. 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 before V4R2, use the Change Telnet Attribute (CHGTELNA) command.

The following examples show 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 IBM i Access job might appear inactive to the system if the user is performing PC functions, such as editing a document, without interacting with the system.

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

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 8. Possible values for the QINACTITV system value:

Value	Description
*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 Inactive Job Time-Out Message Queue (QINACTMSGQ) system value specifies what action the system takes when the inactive job time-out interval for a job has been reached.

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 9. Possible values for QINACTMSGQ system value:

Value	Description
*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.

Table 9. Possible values for QINACTMSGQ system value: (continued)	
Value	**Description**
*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 40 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 topic describes group jobs and secondary jobs.	

Recommended value: *DSCJOB is recommended unless your users run IBM i Access jobs. Using *DSCJOB when some IBM i 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 IBM i Access licensed program. The CL Programming topic 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 IBM i 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 Limit Device Sessions (QLMTDEVSSN) system value specifies whether the number of device sessions allowed for a user is limited.

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.

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 10. Possible values for the QLMTDEVSSN system value:	
Value	**Description**
0	The user is not limited to a specific number of device sessions.
1	The user is limited to a single device session.

Table 10. Possible values for the QLMTDEVSSN system value: (continued)	
Value	**Description**
2 - 9	The user is limited to the specified number of device sessions.

Recommended value: 1 (Yes) is recommended 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 Limit Security Officer (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 202 provides more information about the authority required to sign on at a workstation.

You can always sign on at the console with the QSECOFR, QSRV, and QSRVBAS profiles, no matter how the QLMTSECOFR value is set.

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 11. Possible values for the QLMTSECOFR system value:	
Value	**Description**
1	A user with *ALLOBJ or *SERVICE special authority can sign on at a workstation only if that user is specifically authorized (that is, given *CHANGE authority) to the workstation or if user profile QSECOFR is authorized (given *CHANGE authority) to the workstation. This authority cannot come from public authority.
0	Users with *ALLOBJ or *SERVICE special authority can sign on at any workstation 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 Maximum Sign-On Attempts (QMAXSIGN) system value controls the number of consecutive sign-on or password verification attempts that are not correct by local and remote users.

Incorrect sign-on or password verification 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 or password verification attempts is reached, the QMAXSGNACN system value is used to determine the action to be taken. A CPF1393 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.

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 12. Possible values for the QMAXSIGN system value:	
Value	**Description**
3	A user can try a maximum of 3 sign-on or password verification attempts.
*NOMAX	The system allows an unlimited number of incorrect sign-on or password verification 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 or password verification attempts is three. Typically, 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 Action When Sign-On Attempts Reached (QMAXSGNACN) system value determines what the system does when the maximum number of sign-on or password verification attempts is reached at a workstation.

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 13. Possible values for the QMAXSGNACN system value:	
Value	**Description**
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 or password verification 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)

The Retain Server Security (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 IBM i user profile password.

The recommended value for QRETSVRSEC is 1.

Application failure will occur when QRETSVRSEC is set to 0 because many web servers, IBM i code, and applications require data that is encryptable and decryptable. When QRETSVRSEC is set to 0, storage of this encryptable and decryptable data is not allowed. QRETSVRSEC was originally implemented to provide a layer of security that is no longer necessary because of the current use of the latest level of hardware protection called Hardware Storage Protection (HSP). The internal objects that are used to store the encryptable and decryptable data are created with public authority of *EXCLUDE and are protected with latest level of HSP, which provides the strongest level of protection available on the Power hardware. Only operating system programs can access these objects directly, users must use defined interfaces such as APIs .

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 re-enables 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 Clear Server Security Data (CLRSVRSEC) command. If you have many user profiles or validation lists on your system the **CLRSVRSEC** command might 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.

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 14. Possible values for the QRETSVRSEC system value:

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

Recommended value: 1

Related concepts

Using validation lists

Validation list objects provide a method for applications to securely store user-authentication information.

Remote power-on and restart (QRMTIPL)

One part of your system security plan is to determine whether you will allow remote users to power-on and restart the system. The Remote power-on and restart (QRMTIPL) system value provides you the ability to start the remote system by using your telephone and a modem or the SPCN signal.

When QRMTIPL is set to 1 (Yes), any telephone call causes the system to restart. Even though this system value deals with restart options of your system, it has security implications. Obviously you do not want someone inadvertently restarting your systems. However, if you use a remote system to administer your system you will need to allow remote restart.

Table 15. Possible values for the remote power-on and restart system value (QRMTIPL)	
Value	**Description**
0	Do not allow remote power-on and restart
1	Allow remote power-on and restart

Related information

Restart system values: Allow remote power-on and restart

Remote Sign-On Control (QRMTSIGN)

The Remote Sign-On Control (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 IBM i Access licensed program, and TELNET access.

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 16. Possible values for the QRMTSIGN system value:	
Value	**Description**
*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 can 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. Telnet uses a value of *VERIFY when *SAMEPRF is specified. The sign-on display appears for Telnet and 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. Telnet displays the sign-on display when the security information provided is not valid. If the target system has a QSECURITY value of 10, any automatic sign-on request is allowed. The sign-on display appears for Telnet and pass-through attempts not requesting automatic sign-on.
*REJECT	No remote sign-on is permitted.
	Telnet uses a value of *FRCSIGNON when *REJECT is specified and displays the sign-on display, Pass-through will reject the session request.
program-name library-name	The program specified runs at the start and end of every pass-through session. Telnet uses a value of *FRCSIGNON when this program is specified.

Recommended value: *REJECT is recommended if you do not want to allow any pass-through or IBM i Access access. If you do allow pass-through or IBM i Access access, use *FRCSIGNON or *SAMEPRF.

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

Scan File Systems (QSCANFS)

The Scan File Systems (QSCANFS) system value allows you the option to specify the integrated file system in which objects will be scanned.

For example, you can use this option to scan for a virus. Integrated file system scanning is enabled when exit programs are registered with any of the integrated file system scan-related exit points. The QSCANFS system value specifies the integrated file systems in which objects will be scanned when exit programs are registered with any of the integrated file system scan-related exit points.

The integrated file system scan-related exit points are:

- QIBM_QP0L_SCAN_OPEN — Integrated file system scan on open exit.
- QIBM_QP0L_SCAN_CLOSE — Integrated file system scan on close exit.

For more information about integrated file systems, see the Integrated file system topic.

Table 17. Possible values for the QSCANFS system value:

Value	Description
*NONE	No integrated file system objects will be scanned.
*ROOTOPNUD	Objects of type *STMF that are in *TYPE2 directories in the "root" (/), QOpenSys, and user-defined file systems will be scanned.

Recommended value: The recommended value is *ROOTOPNUD so that the "root" (/), QOpenSys and user-defined file systems are scanned when anyone registers exit programs with the integrated file system scan-related exit points.

Related reference

Scan File Systems Control (QSCANFSCTL)

The Scan File Systems Control (QSCANFSCTL) system value controls the integrated file system scanning that is enabled when exit programs are registered with any of the integrated file system scan-related exit points.

Related information

*TYPE2 directories

Scan File Systems Control (QSCANFSCTL)

The Scan File Systems Control (QSCANFSCTL) system value controls the integrated file system scanning that is enabled when exit programs are registered with any of the integrated file system scan-related exit points.

QSCANFSCTL works with the scan file systems system value to provide granular controls on how and what is scanned in the integrated file system. You can choose different scanning options or you can select to use default scan options. Also, you can select several scan options which control how and what the registered exit programs will scan. These options are described in following table:

Table 18. Possible values for the QSCANFSCTL system value:

Value	Description
*NONE	No controls are being specified for the integrated file system scan-related exit points.

Table 18. Possible values for the QSCANFSCTL system value: (continued)	
Value	**Description**
*ERRFAIL	If there are errors when calling the exit program (for example, program not found or the exit program signals an error), the system will fail the request which triggered the exit program call. If this is not specified, the system will skip the exit program and treat it as if the object was not scanned.
*FSVRONLY	Only accesses through the file servers will be scanned. For example, accesses through Network File System will be scanned as well as other file server methods. If this is not specified, all accesses will be scanned.
*NOFAILCLO	The system will not fail the close requests with an indication of scan failure, even if the object failed a scan which was done as part of the close processing. Also, this value will override the *ERRFAIL specification for the close processing, but not for any other scan-related exit points.
*NOPOSTRST	After objects are restored, they will not be scanned just because they were restored. If the object attribute is that "the object will not be scanned", the object will not be scanned at any time. If the object attribute is that "the object will be scanned only if it has been modified since the last time it was scanned", the object will only be scanned if it is modified after being restored. If *NOPOSTRST is not specified, objects will be scanned at least once after being restored. If the object attribute is that "the object will not be scanned", the object will be scanned once after being restored. If the object attribute is that "the object will be scanned only if it has been modified since the last time it was scanned", the object will be scanned after being restored because the restore will be treated as a modification to the object. In general, it may be dangerous to restore objects without scanning them at least once. It is best to use this option only when you know that the objects were scanned before they were saved or they came from a trusted source.
*NOWRTUPG	The system will not attempt to upgrade the access for the scan descriptor passed to the exit program to include write access. If this is not specified, the system will attempt to do the write access upgrade.
*USEOCOATR	The system will use the specification of the "object change only" attribute to only scan the object if it has been modified (not also because scan software has indicated an update). If this is not specified, this "object change only" attribute will not be used, and the object will be scanned after it is modified and when scan software indicates an update.

Recommended value: If you want the most restrictive values specified for integrated file system scanning, then the recommended settings are *ERRFAIL and *NOWRTUPG. This ensures that any failure from the scan exit programs prevent the associated operations, as well as not give the exit program additional access levels. However, the *NONE value is a good option for most users. When installing code that is shipped from a trusted source, it is recommended that *NOPOSTRST be specified during that install time period.

Related reference

Scan File Systems (QSCANFS)

The Scan File Systems (QSCANFS) system value allows you the option to specify the integrated file system in which objects will be scanned.

Share Memory Control (QSHRMEMCTL)

The Share Memory Control (QSHRMEMCTL) system value defines which users are allowed to use shared memory or mapped memory that has write capability.

Your environment may contain applications, each running different jobs, but sharing pointers within these applications. Using these APIs provides for better application performance and streamlines the application development by allowing shared memory and stream files among these different applications and jobs. However, use of these APIs might potentially pose a risk to your system and assets. A programmer can have write access and can add, change, and delete entries in the shared memory or stream file.

To change this system value, users must have *ALLOBJ and *SECADM special authorities. A change to this system value takes effect immediately.

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 19. Possible values for the QSHRMEMCTL system value:

Value	Description
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 Use Adopted Authority (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.

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 20. Possible values for the QUSEADPAUT system value:

Value	Description
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: • The user does not have authority to the specified authorization list. • There are no other errors when the program or service program is created.
*NONE[1]	All users can create, change, or update programs and service programs to use the authority of the program which called them if the user has the necessary authority to the program or service program.

1
*NONE indicates that no authorization list is used and by default all users will be allowed to access programs that use adopted authority.

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

This topic introduces the security-related system values on your IBM i operating system.

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

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

The following information 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
Disconnected job time-out interval

Note: This system value is also discussed in the Jobs system values: Time-out interval for disconnected jobs topic.

QRMTSRVATR
Remote service attribute

QSSLCSL
Transport Layer Security (TLS) cipher specification list

QSSLCSLCTL
Transport Layer Security (TLS) cipher control

QSSLPCL
Transport Layer Security (TLS) protocols

Related concepts

Validation of programs being restored
When a program is created, the 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.

Automatic Device Configuration (QAUTOCFG)

The Automatic Device Configuration (QAUTOCFG) system value automatically configures locally attached devices. The value specifies whether devices that are added to the system are configured automatically.

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 21. Possible values for the QAUTOCFG system value:

Value	Description
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 Automatic Configuration of Virtual Devices (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.

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 22. Possible values for the QAUTOVRT system value:	
Value	**Description**
0	No virtual devices are created automatically.
number-of- virtual- devices	Specify a value 1 through 32500. 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

Related information
TCP/IP setup

Device Recovery Action (QDEVRCYACN)

The Device Recovery Action (QDEVRCYACN) system value specifies what action to take when an I/O error occurs for an interactive job's workstation.

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 23. Possible values for the QDEVRCYACN system value:	
Value	**Description**
*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 effect 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 job performs the next I/O operation. In an LAN/WAN environment, this allows one device to disconnect and another to connect, using the same address, before the next I/O operation for the job occurs. The job can recover from the I/O error message and continue running to the second device. To avoid this, specify a device recovery action of *DSCENDRQS, *ENDJOB, or *ENDJOBNOLIST. 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.

Disconnected Job Time-Out Interval (QDSCJOBITV)

The Disconnected Job Time-Out Interval (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.

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 24. Possible values for the QDSCJOBITV system value:

Value	Description
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)

The Remote Service Attribute (QRMTSRVATR) controls the remote system service problem analysis ability. The value allows the system to be analyzed remotely.

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

The values allowed for the QRMTSRVATR system value are:

Table 25. Possible values for the QRMTSRVATR system value:

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

Recommended value: 0

Transport Layer Security (TLS) cipher specification list (QSSLCSL)

The Transport Layer Security cipher specification list (QSSLCSL) system value determines the specific cipher suites supported by System TLS. Applications can negotiate secure sessions with only a cipher suite that is listed in QSSLCSL. No matter what an application does with code or configuration, it cannot negotiate secure sessions with a cipher suite if it is not listed in QSSLCSL. Individual application configuration determines which of the enabled cipher suites are used for that application.

System TLS uses the sequence of the values in QSSLCSL to determine the order of the System TLS default cipher specification list. You can refer to the Cipher suite configuration in the TLS topic for additional details on displaying and configuring the default cipher specification list.

A cipher suite cannot be added to QSSLCSL if the required TLS protocol value for the cipher suite is not set for the QSSLPCL (TLS protocol list) system value.

The values of the QSSLCSL system value are read-only unless the TLS cipher control (QSSLCSLCTL) system value is set to *USRDFN.

The values allowed for the QSSLCSL system value are as follows:

- *AES_128_GCM_SHA256

- *AES_256_GCM_SHA384
- *CHACHA20_POLY1305_SHA256
- *RSA_AES_128_GCM_SHA256
- *RSA_AES_256_GCM_SHA384
- *ECDHE_ECDSA_NULL_SHA
- *ECDHE_ECDSA_RC4_128_SHA
- *ECDHE_ECDSA_3DES_EDE_CBC_SHA
- *ECDHE_RSA_NULL_SHA
- *ECDHE_RSA_RC4_128_SHA
- *ECDHE_RSA_3DES_EDE_CBC_SHA
- *ECDHE_ECDSA_AES_128_CBC_SHA256
- *ECDHE_ECDSA_AES_256_CBC_SHA384
- *ECDHE_RSA_AES_128_CBC_SHA256
- *ECDHE_RSA_AES_256_CBC_SHA384
- *ECDHE_ECDSA_AES_128_GCM_SHA256
- *ECDHE_ECDSA_AES_256_GCM_SHA384
- *ECDHE_RSA_AES_128_GCM_SHA256
- *ECDHE_RSA_AES_256_GCM_SHA384
- *ECDHE_ECDSA_CHACHA20_POLY1305_SHA256
- *ECDHE_RSA_CHACHA20_POLY1305_SHA256
- *RSA_AES_128_CBC_SHA256
- *RSA_AES_128_CBC_SHA
- *RSA_AES_256_CBC_SHA256
- *RSA_AES_256_CBC_SHA
- *RSA_3DES_EDE_CBC_SHA
- *RSA_RC4_128_SHA
- *RSA_RC4_128_MD5
- *RSA_DES_CBC_SHA
- *RSA_EXPORT_RC2_CBC_40_MD5
- *RSA_EXPORT_RC4_40_MD5
- *RSA_NULL_SHA256
- *RSA_NULL_SHA
- *RSA_NULL_MD5

Note: You must have *IOSYSCFG, *ALLOBJ, and *SECADM special authorities to change this system value.

You can refer to the Transport Layer Security cipher specification list topic in the System values topic collection for more information about the shipped values.

Related information

Security system values: Transport Layer Security cipher specification list
System TLS System Level Settings

Transport Layer Security (TLS) cipher control (QSSLCSLCTL)

The Transport Layer Security cipher control (QSSLCSLCTL) system value specifies whether the system or the user controls the Transport Layer Security cipher specification list (QSSLCSL) system value.

The values allowed for the QSSLCSLCTL system value are as follows:

- *OPSYS
- *USRDFN

Note: You must have *IOSYSCFG, *ALLOBJ, and *SECADM special authorities to change this system value.

You can refer to the Transport Layer Security cipher control topic in the System values topic collection for more information about the shipped values.

Related information
Security system values: Transport Layer Security cipher control

Transport Layer Security (TLS) protocols (QSSLPCL)

The Transport Layer Security protocols (QSSLPCL) system value specifies the Transport Layer Security (TLS) protocols supported by the System TLS.

The values allowed for the QSSLPCL system value are as follows:

- *OPSYS
- *TLSV1.3
- *TLSV1.2
- *TLSV1.1
- *TLSV1
- *SSLV3

Note: You must have *IOSYSCFG, *ALLOBJ, and *SECADM special authorities to change this system value.

You can refer to the Transport Layer Security protocols topic in the System values topic collection for more information about the shipped values.

Related information
Security system values: Transport Layer Security protocols
System TLS System Level Settings

Security-related restore system values

This topic introduces the security-related restore system values on your IBM i operating system.

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

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

The following information are descriptions of system values that relate to restoring security-related objects on the system which should be considered when restoring objects as well. See Table 18 on page 34 for more information about the QSCANFSCTL *NOPOSTRST system value.

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.

Related concepts

Restoring programs
Restoring programs to your system that are obtained from an unknown source poses a security exposure. This topic provides information about the factors that should be taken into consideration when restoring programs.

Verify Object on Restore (QVFYOBJRST)

The Verify Object on Restore (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 correct 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 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 objects with security-sensitive attributes can be restored.

If Digital Certificate Manager (IBM i 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.

Program, service program and module objects that are created or converted on a system with a release before V6R1 are treated as unsigned when they are restored to a V6R1 or later system. Likewise, program, service program and module objects that are created or converted on a V6R1 or later release are treated as unsigned when they are restored to a system before V6R1.

A change to this system value takes effect immediately.

Notes:

1. This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.
2. Objects that have the system-state attribute and objects that have the inherit-state attribute are required to have a valid signature from a system-trusted source. Objects in Licensed Internal Code PTFs are also required to have a valid signature from a system-trusted source. If these objects do not have a valid signature, they cannot be restored, regardless of the value of the QVFYOBJRST system value.

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 IBM i operating system.

Table 26. Possible values for the QVFYOBJRST system value:	
Value	**Description**
1	Do not verify signatures on restore. Restore all user-state objects regardless of their signature. Do not use this value 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. Use this value only if certain objects that you want to restore contain signatures that are not valid. In general, it is not recommended to restore objects with signatures that are not valid on your system.
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. Use this value 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. Use this value only if certain objects that you want to restore contain signatures that are not valid, but you do not want the possibility of unsigned objects being restored. In general, it is not recommended 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

Some commands use a signature that does not include 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 like, 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)

The Force Conversion on Restore (QFRCCVNRST) system value can force the conversion of some object types during a restore. This system value can also prevent some objects from being restored.

The QFRCCVNRST system value specifies whether to convert the following object types during a restore:

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

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 objects with security-sensitive attributes can be restored.

If Digital Certificate Manager (IBM ioption 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 QFRCCVNRST system value during a restore operation.

Program, service program and module objects that are created or converted on a system with a release before V6R1 are treated as unsigned when they are restored to a V6R1 or later system. Likewise, program, service program and module objects that are created or converted on a V6R1 or later release are treated as unsigned when they are restored to a system before V6R1.

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.

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

The following table summarizes the allowed values for QFRCCVNRST:

Table 27. QFRCCVNRST values

Value	Description
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 the current machine, 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 or on the current machine will be converted.

Table 27. QFRCCVNRST values (continued)	
Value	**Description**
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) that have validation errors, are suspected of having been tampered with, or require conversion to be used on the current version of the operating system or on the current machine 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 that have validation errors, are suspected of having been tampered with, or require conversion to be used on the current version of the operating system or on the current machine 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 Allow Restoring of Security-Sensitive Objects (QALWOBJRST) system value determines whether objects that are security-sensitive may be restored to your system.

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 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 objects with security-sensitive attributes can be restored. You can use it to prevent anyone from restoring a system state object or an object that adopts authority.

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 IBM i 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.

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

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

Table 28. Possible values for the QALWOBJRST system value:

Value	Description
*ALL	Any object can be restored to your system by a user with the correct authority.
*NONE	Security-sensitive objects, such as system state programs or programs that adopt authority, cannot be restored to the system.
*ALWSYSSTT	System and inherit state objects can be restored to the system.
*ALWPGMADP	Objects that adopt authority can 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 might need to set the QALWOBJRST system value to *ALWPGMADP.

System values that apply to passwords

This topic describes the system values that apply to 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.

If the QPWDRULES system value contains any value other than *PWDSYSVAL, the QPWDLMTAJC, QPWDLMTCHR, QPWDLMTREP, QPWDMAXLEN, QPWDMINLEN, QPWDPOSDIF, and QPWDRQDDGT system values are ignored when a new password is checked to see if it is formed correctly.

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

Note:
Changes take effect immediately (except for QPWDLVL). IPL is not required.

The system values control passwords:

QPWDCHGBLK
Block password change

QPWDEXPITV
Expiration interval

QPWDEXPWRN
Password expiration warning

QPWDLVL
Password level

QPWDLMTCHR
Restricted characters

QPWDLMTAJC
Restrict adjacent characters

QPWDLMTREP
Restrict repeating characters

QPWDMINLEN
Minimum length

QPWDMAXLEN
Maximum length

QPWDPOSDIF
Character position difference

QPWDRQDDIF
Required difference

QPWDRQDDGT
Require numeric character

QPWDRULES
Password rules

QPWDVLDPGM
Password validation program

The password-composition system values are always enforced when the password is changed using the **CHGPWD** command, the ASSIST menu option to change a password, or the QSYCHGPW application programming interface (API). The password rules are enforced when using the **CRTUSRPRF** or **CHGUSRPRF** command only when the QPWDRULES system values has the *ALLCRTCHG value specified. If *ALLCRTCHG is not specified in QPWDRULES, then a password that does not meet the currently defined password composition rules can be set for a user by using the CRTUSRPRF or CHGUSRPRF commands. For this scenario where the password does not meet the password rules, the Change Profile (CP) security audit record contains an indication that the password for this user does not conform to the password composition system value rules. The Change Profile (CP) audit record is sent if security auditing is on and *SECURITY actions are being audited, see Chapter 9, “Auditing security on IBM i,” on page 259 for instructions on activating security auditing.

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 in any of the following conditions.

- The Password Rules (QPWDRULES) system value has a value of *PWDSYSVAL and the Password Minimum Length (QPWDMINLEN) system value has a value other than 1.
- The Password Rules (QPWDRULES) system value has a value of *PWDSYSVAL and the Password Maximum Length (QPWDMAXLEN) system value has a value other than 10.
- The Password Rules (QPWDRULES) system value has a value of *PWDSYSVAL and you change any of the other password-control system values from the defaults.

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.

Related information

System values: Password overview

Block Password Change (QPWDCHGBLK)

The Block Password Change (QPWDCHGBLK) system value specifies the time period during which a password is blocked from being changed after the prior successful password change operation.

A change to this system value takes effect immediately.

Note: This system value is a restricted value. Refer to the Security System Values topic for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 29. Possible values for the QPWDCHGBLK system value:

Value	Description
*NONE	The password can be changed at any time.
1 - 99	A password cannot be changed within the specified number of hours after the prior successful password changed operation.

Password Expiration Interval (QPWDEXPITV)

The Password Expiration Interval (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 . . . . . . . . . . . . . :   10/30/99  14:15:00

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

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 30. Possible values for the QPWDEXPITV system value:

Value	Description
*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 Expiration Warning (QPWDEXPWRN)

The Password Expiration Warning (QPWDEXPWRN) system value specifies the number of days before a password expiration to begin displaying the password expiration warning messages when a user signs on.

A change to this system value takes effect immediately.

Note: This system value is a restricted value. Refer to the Security System Values topic for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 31. Possible values for the QPWDEXPWRN system value:	
Value	**Description**
7	Specifies that the password expiration warning message should start to be displayed 7 days before the password expiration.
1 - 99	Specifies the number of days before the password expiration to begin displaying the password expiration warning message.

Recommended value: 14 (days)

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, review the section Planning password level changes.

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 32. Possible values for the QPWDLVL system value:	
Value	**Description**
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 underline. • QPWDLVL 0 should be used if your system communicates with other IBM i platforms 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 IBM i Support for Windows Network Neighborhood (IBM i 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.

Table 32. Possible values for the QPWDLVL system value: (continued)	
Value	**Description**
1	QPWDLVL 1 is the equivalent support of QPWDLVL 0 with the following exception: IBM i NetServer LAN manager passwords for Windows 95/98/ME clients will be removed from the system. The LAN manager password is used to communicate with IBM i Support for Windows Network Neighborhood (IBM i NetServer) product and only affects Windows 95/98/ME clients. The LAN manager passwords have been disabled by Windows since Vista so removing them will not affect current versions of Windows. Unless the Windows 95/98/ME clients are configured to use NTLMv2 passwords, you cannot use QPWDLVL value 1 to connect those clients to the IBM i NetServer product. QPWDLVL 1 improves the security of IBM i platforms by removing all IBM i NetServer LAN manager passwords from the system.
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 IBM i Support for Windows Network Neighborhood (IBM i NetServer) product as long as your password is 1-14 characters in length. • QPWDLVL 2 cannot be used if your system communicates with other IBM i platforms 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 IBM i platforms 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 IBM i Support for Windows Network Neighborhood (IBM i NetServer) product. At level 3 the IBM i NetServer LAN manager passwords for Windows 95/98/ME clients will be removed from the system. The LAN manager password is used to communicate with IBM i Support for Windows Network Neighborhood (IBM i NetServer) product and only affects Windows 95/98/ME clients. The LAN manager passwords have been disabled by Windows since Vista so removing them will not affect current versions of Windows. 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 Display Security Attributes (**DSPSECA**) command .

Password Encryption and Storage on IBM i

IBM i password encryption does not use a "hardcoded" encryption key in either of the password encryption algorithms so there is no key that needs to be stored or protected. The encryption algorithms use the USERID and the PASSWORD itself in the encryption algorithm. Before actually encrypting and/or hashing (depending on the setting of the QPWDLVL system value), there are a few additional steps that are performed to essentially drop off a few of the bits that make up the clear text password followed by an "exclusive or" operation on the password string (this helps protect the password). This password string is then used to encrypt or hash the user ID in order to create the encrypted password. Since the password itself becomes the key, things are very secure as a key does not need to be stored anywhere on the system. When it is time to authenticate a user, the system takes the clear text password that the user entered (on the signon screen, etc.) and runs the same algorithm, then compares that encrypted result with the encrypted result that was created and saved when the password was changed. There is never a comparison that is done with the clear text password itself since the encryption algorithms are both one-way, meaning you can never decrypt and get back the clear text password.

The user profile passwords are stored in an internal control block that is protected with the strongest mechanism available to the IBM i operating system running on the Power hardware. A capability that is called Hardware Storage Protection (HSP) is used to protect the control block. The HSP capability is protection that is built into the Power hardware and enforced by the hardware itself. The HSP value that is used is called "no access from user state" and "protect at all security levels". This HSP protection value keeps all user level code out of the control block (no read or write access) but allows the operating system to read/write the control block. This protection is always activated as the control block is protected at all security levels. If user level code tries to access the control block, the hardware would send an exception and the Licensed Internal Code would send an error to the user level code (and access would be denied).

If someone has the encrypted password could they decrypt it to get the clear text password?

No, but a brute force attack is possible, basically running all potential passwords through the algorithm and comparing the encrypted results. So it is important to protect your SAVSYS and SAVSECDTA tapes and data by using encrypted backup with tape hardware capable of encryption. The operating system protects the passwords by storing them in an internal control block that is protected with the strongest mechanism available to the operating system on the Power hardware. HSP is used to protect the control block. But the passwords are saved on media during a SAVSYS and SAVSECDTA so the media needs to be protected (encrypted backup and physical security).

One thing to be aware of is that the system has two IBM i APIs, set encrypted password (QSYSUPWD) and retrieve encrypted password (QSYRUPWD) that were implemented to allow the High Availability (HA) business partners the ability to move user profile changes from the production machine to the target side backup server. These APIs allow the retrieve and set of the encrypted password for a user profile but the APIs are only callable by a security officer (*ALLOBJ and *SECADM special authority required). These APIs do return the encrypted password string so this data and the use of the API need to be well controlled. The HA partners use these APIs to move the password to the target server when a password changes on the production server in order to keep the password change in sync. The encrypted password string, and other information, returned by the QSYRUPWD API have a cyclic redundancy check (CRC) created to ensure the password itself is not modified (either intentionally or accidentally) when being moved from system to system. The CRC is checked by the QSYSUPWD API to ensure that the string is the same as when it was returned by QSYRUPWD. This CRC does not provide any protection for the encrypted password itself, it just ensures that the string isn't changed before setting the password on the target server. To protect the encrypted password in the HA environment (along with all data flowing from source to target), an encrypted session between the source and target system is recommended.

Minimum Length of Passwords (QPWDMINLEN)

The Minimum Length of Passwords (QPWDMINLEN) system value controls the minimum number of characters in a password.

Notes:

1. This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.
2. If the QPWDRULES system value is any value other than *PWDSYSVAL, this system value cannot be changed and its value will be ignored when new passwords are checked to see if they are formed correctly.

Table 33. Possible values for the QPWDMINLEN system value:

Value	Description
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: 7 is recommended to prevent users from assigning passwords that are easily guessed, such as initials or a single character.

Maximum Length of Passwords (QPWDMAXLEN)

The Maximum Length of Passwords (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 need 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.

Notes:

1. This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.
2. If the QPWDRULES system value specifies any value other than *PWDSYSVAL, this system value cannot be changed and its value will be ignored when new passwords are checked to see if they are formed correctly.

Table 34. Possible values for the QPWDMAXLEN system value:

Value	Description
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 higher.

Recommended value: 10 when QPWDLVL is 0 or 1. 128 when QPWDLVL is 2 or higher.

Required Difference in Passwords (QPWDRQDDIF)

The Required Difference in Passwords (QPWDRQDDIF) system value controls whether the password must be different from previous passwords.

This value provides additional security by preventing users from specifying passwords that were 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. This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 35. 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 Required Difference in Passwords (QPWDRQDDIF) system value and the Password Expiration Interval (QPWDEXPITV) 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 Restricted Characters for Passwords (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.

Notes:

1. This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.
2. If the QPWDRULES system value specifies any value other than *PWDSYSVAL, this system value cannot be changed and its value will be ignored when new passwords are checked to see if they are formed correctly.

Table 36. Possible values for the QPWDLMTCHR system value:	
Value	**Description**
*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 underline (_).

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

Restriction of Consecutive Digits for Passwords (QPWDLMTAJC)

The Restriction of Consecutive Digits for Passwords (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.

Notes:

1. This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.
2. If the QPWDRULES system value specifies any value other than *PWDSYSVAL, this system value cannot be changed and its value will be ignored when new passwords are checked to see if they are formed correctly.

Table 37. Possible values for the QPWDLMTAJC system value:	
Value	**Description**
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 Restriction of Repeated Characters for Passwords (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'.

Notes:

1. This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.
2. If the QPWDRULES system value specifies any value other than *PWDSYSVAL, this system value cannot be changed and its value will be ignored when new passwords are checked to see if they are formed correctly.

Table 38. Possible values for the QPWDLMTREP system value:	
Value	**Description**
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.

Table 38. Possible values for the QPWDLMTREP system value: (continued)	
Value	**Description**
2	The same character cannot be used consecutively in a password.

Table 39 on page 56 shows examples of what passwords are allowed based on the QPWDLMTREP system value.

Table 39. 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 40. 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 Character Position Difference for Passwords (QPWDPOSDIF) system value controls each position in a new password.

This system value 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'.

Notes:

1. This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.
2. If the QPWDRULES system value specifies any value other than *PWDSYSVAL, this system value cannot be changed and its value will be ignored when new passwords are checked to see if they are formed correctly.

Table 41. Possible values for the QPWDPOSDIF system value:	
Value	**Description**
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 Requirement for Numeric Character in Passwords (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.

Notes:

1. This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.
2. If the QPWDRULES system value specifies any value other than *PWDSYSVAL, this system value cannot be changed and its value will be ignored when new passwords are checked to see if they are formed correctly.

Table 42. Possible values for the QPWDRQDDGT system value:

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

Recommended value: 1

Password Rules (QPWDRULES)

The Password Rules (QPWDRULES) system value specifies the rules used to check whether a password is formed correctly. You can specify more than one value for the QPWDRULES system value, unless you specify *PWDSYSVAL.

Changes made to this system value take effect the next time a password is changed.

Note: This system value is a restricted value. Refer to the Security System Values topic for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 43. Possible values for the QPWDRULES system value:

Value	Description
*PWDSYSVAL	This value specifies that the QPWDRULES system value is ignored and the other password system values are used to check whether a password is formed correctly. These other password system values include QPWDLMTAJC, QPWDLMTCHR, QPWDLMTREP, QPWDMAXLEN, QPWDMINLEN, QPWDPOSDIF, and QPWDQDDGT. **Note:** If any value other than *PWDSYSVAL is specified for QPWDRULES, the QPWDLMTAJC, QPWDLMTCHR, QPWDLMTREP, QPWDMAXLEN, QPWDMINLEN, QPWDPOSDIF, and QPWDRQDDGT system values are ignored when a new password is checked to see if it is formed correctly. In addition, any attempt to change these system values will be rejected as long as the QPWDRULES system value contains a value other than *PWDSYSVAL.

Table 43. Possible values for the QPWDRULES system value: (continued)

Value	Description
*ALLCRTCHG	Enforce all password composition rules defined in the QPWDRULES system value when setting a password via the Create User Profile (CRTUSRPRF) command or the Change User Profile (CHGUSRPRF) command. CRTUSRPRF and CHGUSRPRF validation programs registered for the QIBM_QSY_VLD_PASSWRD exit point, format VLDP0200, will be called to validate the password after the password composition rules have been checked. **Note:** Password composition rules are always enforced when using the Change Password (CHGPWD) command and the Change User Password (QSYCHGPW) API, regardless of whether or not *ALLCRTCHG is specified.

*CHRLMTAJC	The value specifies that a password cannot contain 2 or more occurrences of the same character that are positioned adjacent to each other. This value performs the same function as specifying a value of 2 for the QPWDLMTREP system value. If the *CHRLMTREP value was specified, this value cannot be specified.
	
	Examples:
	`Better.test  not valid - tt`
	`fix11bugs  not valid - 11`
	`@12/A78  valid`
	`A1234A1234  valid`

*CHRLMTREP	The value specifies that a password cannot contain 2 or more occurrences of the same character. This value performs the same function as specifying a value of 1 for the QPWDLMTREP system value. If the *CHRLMTAJC value was specified, this value cannot be specified.
	
	Examples:
	`John.Jones  not valid - J o n`
	`THISONEOK  not valid - O`
	`@12/A78  valid`
	`AaCcEeFfGg  valid`

*DGTLMTAJC	The value specifies that a password cannot contain 2 or more adjacent digit characters.
	
	Examples:
	`@12/A78  not valid`
	`!@#$%a1234.  not valid`
	`THISONEOK  valid`
	`A1B2C3DE5  valid`

Table 43. Possible values for the QPWDRULES system value: (continued)

Value	Description

*DGTLMTFST	The value specifies that the first character of a password cannot be a digit character. If *LTRLMTFST and *SPCCHRLMTFST values were specified, this value cannot be specified. If the system is operating at password level 0 or 1, the system functions like the *DGTLMTFST value is specified.
	Examples:
	`16ST-SW-Roch  not valid - 1`
	`99BottlesOfBeer  not valid - 9`
	`@12/A78  valid`
	`Allow-this.1  valid`

*DGTLMTLST	The value specifies that the last character of the password cannot be a digit character. If *LTRLMTLST and *SPCCHRLMTLFST values were specified, this value cannot be specified.
	Examples:
	`John.doe12  not valid - 2`
	`@12/A78  not valid - 8`
	`THISONEOK  valid`
	`A1234b123.  valid`

*DGTMAXn	The value specifies the maximum number of digit characters that can occur in the password. The **n** is a number from 0 to 9.
	Only one *DGTMAXn value can be specified. If a *DGTMINn value is also specified, the n value specified for *DGTMAXn must be greater than or equal to the n value specified for *DGTMINn.
	Examples: for *DGTMAX2
	`Q12345678  not valid - 6 digits too many`
	`3-2-1->Go  not valid - 1 digit too many`
	`Rick1  valid`
	`Ed1-Jeff3  valid`

*DGTMINn	The value specifies the minimum number of digit characters that must occur in the password. The **n** is a number from 0 to 9.
	Only one *DGTMINn value can be specified. If a *DGTMAXn value is also specified, the n value specified for *DGTMAXn must be greater than or equal to the n value specified for *DGTMINn.
	Examples: for *DGTMIN3
	`Rick1  not valid - only 1 digit`
	`Ed1-Jeff3  not valid - only 2 digits`
	`3-2-1->Go  valid`
	`Q12345678  valid`

Table 43. Possible values for the QPWDRULES system value: (continued)

Value	Description

*LMTSAMPOS	The same character cannot be used in a position corresponding to the same position in the previous password. This value performs the same function as the QPWDPOSDIF system value.
	
	*LMTSAMPOS will not be enforced when the password is set by the Change User Profile (CHGUSRPRF) command or the Create User Profile (CRTUSRPRF) command since the previous password value is not supplied. It will only be enforced when the password is changed by the Change Password (CHGPWD) command or the Change User Password (QSYCHGPW) API.
	
	Examples: for *LMTSAMPOS when Vote4Me was previous password:
	
	`Victory1 not valid - V in position 1`
	`Mine2love not valid - e in position 4`
	`vOTE-mE valid (case is different)`
	`Allisgood valid`

*LMTPRFNAME	The uppercase password value cannot contain the complete user profile name in consecutive positions.
	
	Examples: for *LMTPRFNAME with profile name is JOHNB:
	
	`bigJOHNB9 not valid - positions 4-8`
	`JohnB78 not valid - positions 1-5`
	`J_ohn_B234 valid`
	`john_b valid`

*LTRLMTAJC	The value specifies a password cannot contain 2 or more adjacent letter characters.
	
	Examples:
	
	`John.Smith not valid`
	`THISONEOK not valid`
	`@12/A78 valid`
	`A1234b1234 valid`

*LTRLMTFST	The value specifies the first character of the password cannot be a letter character. If *DGTLMTFST and *SPCCHRLMTFST values were specified, this value cannot be specified. If the system is operating with a QPWDLVL value of 0 or 1, *LTRLMTFST and *SPCCHRLMTFST cannot both be specified.
	
	Examples:
	
	`John.Smith not valid - J`
	`THISONEOK not valid - T`
	`@12/A78 valid`
	`16ST-SW-Roch valid`

Table 43. Possible values for the QPWDRULES system value: (continued)

Value	Description

*LTRLMTLST	The value specifies the last character of the password cannot be a letter character. If *DGTLMTLST and *SPCCHRLMTLST values were specified, this value cannot be specified
	Examples:
	`John.Smith  not valid - h`
	`1Allow.It  not valid - t`
	`@12/A78  valid`
	`(pay*rate)  valid`

*LTRMAXn	The value specifies the maximum number of letter characters that can occur in the password. The **n** is a number from 0 to 9.
	Only one *LTRMAXn value can be specified. If a *LTRMINn value is also specified, the n value specified for *LTRMAXn must be greater than or equal to the n value specified for *LTRMINn.
	If a *MIXCASEn value is also specified, the n value specified for *LTRMAXn must be greater than or equal to 2 times the n value specified for *MIXCASEn.
	Examples: for *LTRMAX4
	`THISONEOK  not valid - 5 letters too many`
	`John.Smith1  not valid - 5 letters too many`
	`John1423  valid`
	`A1b2.#456  valid`

*LTRMINn	The value specifies the minimum number of letter characters that must occur in the password. The **n** is a number from 0 to 9.
	Only one *LTRMINn value can be specified. If a *LTRMAXn value was specified, the n value specified for *LTRMAXn must be greater than or equal to the n value specified for *LTRMINn.
	Examples: for *LTRMIN2
	`@12/A78  not valid - only 1 letter`
	`!@#$%a1234  not valid - only 1 letter`
	`THISONEOK  valid`
	`A1234b1234  valid`

Table 43. Possible values for the QPWDRULES system value: (continued)	
Value	**Description**
*MAXLENnnn	The value specifies the maximum number of characters in a password. The **nnn** is a number from 1 to 128 (without leading zeros). This value performs the same function as the QPWDMAXLEN system value. If the system is operating at QPWDLVL 0 or 1, the valid range is from 1 to 10. If the system is operating at QPWDLVL 2 or 3, the valid range is from 1 to 128. The nnn value specified must be large enough to accommodate all *MIXCASEn, *DGTMAXn, *LTRMAXn, *SPCCHRMAXn, first and last character restrictions, and non-adjacent character requirements. If *MINLENnnn is also specified, the nnn value specified for *MAXLENnnn must be greater than or equal to the nnn value specified for *MINLENnnn. If no *MAXLENnnn value is specified, a value of *MAXLEN10 is assumed if the system is operating with a QPWDLVL value of 0 or 1 or a value of *MAXLEN128 is assumed if the system is operating with a QPWDLVL value of 2 or 3.
*MINLENnnn	The value specifies the minimum number of characters in a password. The **nnn** is a number from 1 to 128 (without leading zeros). If the system is operating at QPWDLVL 0 or 1, the valid range is from 1 to 10. If the system is operating at QPWDLVL 2 or 3, the valid range is from 1 to 128. If *MAXLENnnn is also specified, the nnn value specified for *MAXLENnnn must be greater than or equal to the nnn value specified for *MINLENnnn. If no *MINLENnnn value is specified, a value of *MINLEN1 is assumed.

Table 43. Possible values for the QPWDRULES system value: (continued)

Value	Description

*MIXCASEn	The value specifies a password must contain at least n uppercase and n lowercase letters. The **n** is a number from 0 to 9. This value is rejected if the system is operating with a QPWDLVL value of 0 or 1 because passwords are required to be uppercase.
	Only one *MIXCASEn value can be specified.
	If a *LTRMAXn value was specified, the n value specified for *LTRMAXn must be greater than or equal to two times the n value specified for *MIXCASEn.
	Examples: for *MIXCASE2
	`@12/A78bC  not valid - missing 1 lowercase`
	`THISONEOK  not valid - missing 2 lowercase`
	`ThisIsOkay  valid`
	`Allow-It  valid`

*REQANY3	The value specifies a password must contain characters from at least three of the following four types of characters.
	• Uppercase letters
	• Lowercase letters
	• Digits
	• Special characters
	When the system is operating with a QPWDLVL of 0 or 1, *REQANY3 has the same effect as if *DGTMIN1, *LTRMIN1, and *SPCCHRMIN1 were all specified.
	Examples:
	`THISONEOK  not valid - only 1 type`
	`@12/-78  not valid - only 2 types`
	`A1234b1234  valid - upper, lower, digit`
	`John.Smith  valid - upper, lower, special`
	`peter(21)  valid - lower, special, digit`

*SPCCHRLMTAJC	The value specifies a password cannot contain 2 or more adjacent (consecutive) special characters. A character is considered as a special character if its equivalent unicode character has the property of not being a letter nor a digit.
	Examples:
	`Big//Box  not valid`
	`this->way  not valid`
	`@12/A78  valid`
	`John.Smith  valid`

Table 43. Possible values for the QPWDRULES system value: (continued)

Value	Description

*SPCCHRLMTFST	The value specifies the first character of the password cannot be a special character. A character is considered as a special character if its equivalent unicode character has the property of not being a letter nor a digit.
	
	If *DGTLMTFST and *LTRLMTFST values were specified, this value cannot be specified. If the system is operating with a QPWDLVL value of 0 or 1, *LTRLMTFST and *SPCCHRLMTFST cannot both be specified.
	
	Examples:
	`(2+2equals4)  not valid - (`
	`#fred/#charlie not valid - #`
	`1Good->one12  valid`
	`A1234b1234  valid`

*SPCCHRLMTLST	The value specifies the last character of the password cannot be a special character. A character is considered as a special character if its equivalent unicode character has the property of not being a letter nor a digit.
	
	If *DGTLMTLST and *LTRLMTLST values were specified, this value cannot be specified.
	
	Examples:
	`A1234b123.  not valid - .`
	`>John.Doe<  not valid - <`
	`THISONEOK  valid`
	`@12/A78  valid`

*SPCCHRMAXn	The value specifies the maximum number of special characters that may occur in the password. The **n** is a number from 0 to 9. A character is considered as a special character if its equivalent unicode character has the property of not being a letter nor a digit.
	
	Only one *SPCCHRMAXn value can be specified. If a *SPCCHRMINn value was specified, the n value specified for *SPCCHRMAXn must be greater than or equal to the n value specified for *SPCCHRMINn.
	
	Examples: for *SPCCHRMAX3
	`@12/A78.b#  not valid - 1 too many`
	`!@#$%a1234  not valid - 2 too many`
	`THISONEOK  valid`
	`A1234b-234  valid`

Table 43. Possible values for the QPWDRULES system value: (continued)	
Value	**Description**

*SPCCHRMINn	The value specifies the minimum number of special characters that must occur in the password. The **n** is a number from 0 to 9. A character is considered as a special character if its equivalent unicode character has the property of not being a letter nor a digit.
	
	Only one *SPCCHRMINn value can be specified. If a *SPCCHRMAXn value was specified, the n value specified for *SPCCHRMAXn must be greater than or equal to the n value specified for *SPCCHRMINn.
	
	Examples: for *SPCCHRMIN4
	
	`Su@us.ibm.com  not valid - 1 too few`
	`123+45=168     not valid - 2 too few`
	`A.B@us.ibm.com valid`
	`(24/8=3)       valid`

Password Approval Program (QPWDVLDPGM)

You can specify the Password Approval Program (QPWDVLDPGM) to control the validation of new passwords.

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.

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

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 44. Possible values for the QPWDVLDPGM system value:	
Value	**Description**
*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, format VLDP0100. 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 validation programs are called by the Change Password (CHGPWD) command or Change Password (QSYCHGPW)

API. The validation programs are called only if the new password has passed all other tests specified in the password-control system values. The validation programs are not called from the Create User Profile (CRTUSRPRF) command or the Change User Profile (CHGUSRPRF) command. CRTUSRPRF and CHGUSRPRF commands call validation programs registered for the QIBM_QSY_VLD_PASSWRD exit point, format VLDP0200, if the QPWDRULES system value contains the value *ALLCRTCHG and if the password has passed all other tests 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 45. 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 about 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 before the password can be changed.

Attention: The current and new password are passed to the validation program without encryption. The validation program can 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 QPWDVLDPGM. 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:

Note: By using the code examples, you agree to the terms of the Chapter 11, “Code license and disclaimer information,” on page 333.

```
/*************************************************/
/* 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,     */
/*  format VLDP0100.                                        */
/*                                                          */
/*                                                          */
/*  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.                                    */
/************************************************************/

PGM   PARM(&EXINPUT &RTN)
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 &INDX2      *DEC 5 0
DCL &INDX3      *DEC 5 0
DCL &UNLEN      *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 *GT &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 *GT &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

Auditing system activity is an important part of system security, as it can help detect system misuse and intrusions. You can use specific systems values to control auditing on the IBM i operating system.

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

Note:
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

QAUDLVL2
Auditing level extension

QCRTOBJAUD
Create default auditing

Auditing Control (QAUDCTL)

The Auditing Control (QAUDCTL) system value determines whether auditing is performed.

This system value functions like an on and off switch for the following operations:

- The QAUDLVL and QAUDLVL2 system values
- The auditing defined for objects using the Change Object Auditing (**CHGOBJAUD**), Change Auditing Value (**CHGAUD**), and Change DLO Auditing (**CHGDLOAUD**) commands
- The auditing defined for users using the Change User Audit (**CHGUSRAUD**) command

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

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

Table 46. Possible values for the QAUDCTL system value

Value	Description
*NONE	No auditing is performed for user actions and objects.
*NOTAVL	This value is displayed to indicate that the system value is unavailable to the user because the user has neither *AUDIT nor *ALLOBJ special authority. You cannot set the system value to this value.
*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 and QAUDLVL2 system values 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 IBM i," on page 259 for more details. You must specify this value with either *OBJAUD or *AUDLVL.

See "Planning security auditing" on page 265 for a complete description of the process for controlling auditing on your system.

Auditing End Action (QAUDENDACN)

The Auditing End Action (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.

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 47. Possible values for the QAUDENDACN system value:

Value	Description
*NOTAVL	This value is displayed to indicate that the system value is not available to the user because the user does not have either *AUDIT or *ALLOBJ special authority. The system value cannot be set to this 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 might cause a lengthy initial program load (IPL) when your system is powered on again.

Auditing Force Level (QAUDFRCLVL)

The Auditing Force Level (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.

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 48. Possible values for the QAUDFRCLVL system value

Value	Description
*NOTAVL	This value is displayed to indicate that the system value is not available to the user because the user does not have either *AUDIT or *ALLOBJ special authority. The system value cannot be set to this value.
*SYS	The system determines when journal entries are written to auxiliary storage based on internal system performance.

Table 48. Possible values for the QAUDFRCLVL system value (continued)	
Value	**Description**
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 effect 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 might impair performance.

Auditing Level (QAUDLVL)

The Auditing Level (QAUDLVL) system value along with the QAUDLVL2 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.

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 49. Possible values for the QAUDLVL system value	
Value	**Description**
*NONE	No events controlled by the QAUDLVL or QAUDLVL2 system values are logged. Events are logged for individual users based on the AUDLVL values of user profiles.
*AUDLVL2	Both QAUDLVL and QAUDLVL2 system values will be used to determine the security actions to be audited.
See QAUDLVL2 system value for additional values.	

Related reference

Planning the auditing of actions

The QAUDCTL (audit control) system value, the QAUDLVL (audit level) system value, the QAUDLVL2 (audit level extension) system value, and the AUDLVL (action auditing) parameter in user profiles work together to control action auditing.

Auditing Level Extension (QAUDLVL2)

The Auditing Level Extension (QAUDLVL2) system value is required when more than sixteen auditing values are needed.

Specifying *AUDLVL2 as one of the values in the QAUDLVL system value will cause the system to also look for auditing values in the QAUDLVL2 system value. You can specify more than one value for the QAUDLVL2 system value, unless you specify *NONE. For the QAUDLVL2 system value to take effect, the QAUDCTL system value must include *AUDLVL and the QAUDLVL system value must include *AUDLVL2.

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 50. Possible values for the QAUDLVL2 system value	
Value	**Description**
*NONE	No auditing values are contained in this system value.

Table 50. Possible values for the QAUDLVL2 system value (continued)	
Value	**Description**
*NOTAVL	This value is displayed to indicate that the system value is not available to the user because the user does not have either *AUDIT or *ALLOBJ special authority. The system value cannot be set to this value.
*ATNEVT	Attention events are logged.
*AUTFAIL	Authority failure events are logged.
*CREATE	Object create operations are logged.
*DELETE	Object delete operations are logged.
*JOBBAS	Job base functions are audited.
*JOBCHGUSR	Changes to a thread's active user profile or its group profiles are audited.
*JOBDTA	Actions that affect a job are logged. *JOBDTA is composed of two values, which are *JOBBAS and *JOBCHGUSR, to enable you to better customize your auditing. If both of the values are specified, you will get the same auditing as if just *JOBDTA is specified.
*NETBAS	Network base functions are audited.
*NETCLU	Cluster and cluster resource group operations are audited.
*NETCMN	Network and communication functions are audited. *NETCMN is composed of several values to allow you to better customize your auditing. The following values make up *NETCMN: *NETBAS *NETCLU *NETFAIL The Mail and DHCP functions from *NETSCK
*NETFAIL	Network failures are audited.
*NETSCK	Socket tasks are audited. **Note:** Telnet server connections are not audited as part of *NETSCK. Use *NETTELSVR along with *NETSCK if Telnet server connections should be audited. **Note:** To audit all TCP and UDP connections in and out of the system specify *NETSCK, *NETUDP, and *NETTELSVR.
*NETSECURE	Secure network connections are audited. **Note:** This implies traffic flowing over the connection is now protected by a security protocol known to the system. The system explicitly audits System SSL/TLS and IPsec from operating system code responsible for creating the secure connection. IPsec entries for UDP are created using the same frequency as defined for *NETUDP. The system implicitly audits some non-operating system implemented security protocols by inspecting application layer data as it flows through the Sockets APIs. **Note:** When *NETTELSVR is also specified, telnet secure network connections are audited.

Table 50. Possible values for the QAUDLVL2 system value (continued)	
Value	**Description**
*NETTELSVR	Telnet Server connections are audited. **Note:** Telnet clients can be configured to retry the connection attempt after an attempt to establish a session is unsuccessful. These Telnet clients will retry indefinitely until the conditions causing the session to fail are eliminated. This can generate a large number of Telnet server audit journal entries. **Note:** To audit all TCP and UDP connections in and out of the system specify *NETSCK, *NETUDP, and *NETTELSVR.
*NETUDP	User Datagram Protocol (UDP) traffic is audited. **Note:** UDP traffic for the same local and remote address and port is audited only once every 12 hours by default. Refer to The IPCONFIG macro for details on how to change the default interval. **Note:** To audit all TCP and UDP connections in and out of the system specify *NETSCK, *NETUDP, and *NETTELSVR.
*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.
*PTFOBJ	Changes to PTF objects are logged.
*PTFOPR	PTF operations are logged.
*SAVRST	Restore operations are logged.
*SECCFG	Security configuration is audited.
*SECDIRSRV	Changes or updates when doing directory service functions are audited.
*SECIPC	Changes to interprocess communications are audited.
*SECNAS	Network authentication service actions are audited.
*SECRUN	Security run time functions are audited.
*SECSCKD	Socket descriptors are audited.

Table 50. Possible values for the QAUDLVL2 system value (continued)	
Value	**Description**
*SECURITY	Security-related functions are logged. *SECURITY is composed of several values to allow you to better customize your auditing. The following values make up *SECURITY: *SECCFG *SECDIRSRV *SECIPC *SECNAS *SECRUN *SECSCKD *SECVFY *SECVLDL
*SECVFY	Use of verification functions are audited.
*SECVLDL	Changes to validation list objects are audited.
*SERVICE	Using service tools is logged.
*SPLFDTA	Actions performed on spooled files are logged.
*SYSMGT	Use of systems management functions is logged.

Related reference

Planning the auditing of actions

The QAUDCTL (audit control) system value, the QAUDLVL (audit level) system value, the QAUDLVL2 (audit level extension) system value, and the AUDLVL (action auditing) parameter in user profiles work together to control action auditing.

Auditing for New Objects (QCRTOBJAUD)

The Auditing for New Objects (QCRTOBJAUD) system value is used to determine the auditing value for a new object, if the create object auditing default for the library or directory 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** or **CHGAUD** command.

Note: This system value is a restricted value. See Security system values for details on how to restrict changes to security system values and a complete list of the restricted system values.

Table 51. Possible values for the QCRTOBJAUD system value:	
Value	**Description**
*NONE	No auditing is done for the object.
*NOTAVL	This value is displayed to indicate that the system value is not available to the user because the user does not have either *AUDIT or *ALLOBJ special authority. The system value cannot be set to this value.
*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 a security relevant change is made to the object.

Table 51. Possible values for the QCRTOBJAUD system value: (continued)	
Value	**Description**
*ALL	An audit record is written for any security relevant action that affects the contents of the object. An audit record is also written if a security relevant change is made to the object.

Recommended value: The value you select depends on the auditing requirements of your installation. "Planning the auditing of object access" on page 296 provides more information about methods for setting up object auditing on your system. You can control the auditing value at the directory level with the CRTOBJAUD parameter on the Make Directory (**CRTDIR**) command, and the *CRTOBJAUD value on the Change Attribute (**CHGATR**) command. You can 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

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:

AD for changes to user auditing

CO for creation of a user profile

CP for changes to users profiles

DO for deletion of a user profile

ZC for changes to a user profile that are not relevant to security

Related concepts
User profiles
On the IBM i operating system, every system user has a user profile.

Roles of the user profile

A user profile contain a user's passwords, the list of special authorities assigned to a user, and the objects the user owns.

A 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. "Default values for user profiles" on page 345 in Appendix B, "IBM-supplied user profiles," on page 345 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 that provides the same authority to a group of users.

A group profile 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 can 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 when you create it by specifying a value in the gid parameter. "Planning group profiles" on page 240 shows an example of setting up a group profile.

User-profile parameter fields

This topic describes detailed information about the parameter fields for user profiles shown on the Create User Profile command prompt.

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

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 following sections show what the user profile fields are called on both the basic assistance level and the intermediate assistance level displays.

Field title
The title of the section shows how the field name appears on the Create User Profile command prompt. The title displays 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 field associated with the parameter.

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 corresponding 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

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.

Add User prompt:
User

CL parameter:
USRPRF

Length:
10

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 ($), underline (_), at (@).

The user profile name cannot begin with a number.

Notes:

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

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 that you use to manage user profiles show the user profiles in alphabetical order by user profile name.
- Avoid using special characters in user profile names. Special characters might cause problems with keyboard mapping for certain workstations or with national language versions of the IBM i licensed program.

One technique for assigning user profile names is to use the first seven characters of the family 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: To easily identify group profiles on the system, use a naming convention. Begin all group profile names with the same characters, such as GRP (for group) or DPT (for department).

Password

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).

Add User prompt:
Password

CL parameter:
PASSWORD

Length:
128

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 sign-on 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 cannot begin with an asterisk character ('*'), and trailing blank characters in the password are removed.

Note: Passwords can be created using double-byte characters. However, a password containing double-byte characters cannot be used to sign on via the system sign-on 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 are always enforced when a user changes a password using the Change Password (CHGPWD) command, the Change password option from the ASSIST menu, or the QSYCHGPW API. The password rules are enforced when using the **CRTUSRPRF** or **CHGUSRPRF** command only when the QPWDRULES system values has the *ALLCRTCHG value specified. If *ALLCRTCHG is not specified in QPWDRULES, then a password that does not meet the currently defined password composition rules can be set for a user via the CRTUSRPRF or CHGUSRPRF commands. For this scenario where the password does not meet the password rules, the Change Profile (CP) security audit record will contain an indication that the password for this user does not conform to the password composition system value rules. The Change Profile (CP) audit record is sent if security auditing is on and *SECURITY actions are being audited, see Chapter 9, "Auditing security on IBM i," on page 259 for instructions on activating security auditing. A user cannot set the password equal to the user profile name using the CHGPWD command, the ASSIST menu, or the QSYCHGPW API in any of the following conditions.

- The QPWDRULES system value is *PWDSYSVAL and the Password Minimum Length (QPWDMINLEN) system value is not 1.
- The QPWDRULES system value is *PWDSYSVAL and the Password Maximum Length (QPWDMAXLEN) system value is not 10.

- The QPWDRULES system value is *PWDSYSVAL and any of the other password composition system values have been changed from the default values.

See the topic "System values that apply to passwords" on page 47 for information about setting the password composition system values.

Table 52. Possible values for PASSWORD:

Value	Description
*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 is 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 correct authority to the user profile.
user- password	A character string (128 characters or less).

Using variant characters in a password can lead to potential issues when IBM i validates passwords

An invariant character has the same code point among all supported IBM i CCSIDs. Examples of invariant characters are A-Z and 0-9, but there are more characters that are also invariant. Using invariant characters in your passwords is a good practice since you will be able to communicate with systems running with different CCSIDs and languages. For more information on invariant characters, see Invariant character set (and its exceptions).

A variant character is one that may translate to a different code point depending on the language and CCSID being used.

For example, compare CCSID 37 and CCSID 277 (Danish):

```
@ in CCSID 37  -> Code point x'7C'
@ in CCSID 277 -> Code point x'80'
Ø in CCSID 277 -> Code point x'7C'
```

Assume a user is running in CCSID 37 and uses the CHGPWD command to set their password to PWD@123. The user now opens a Navigator for i session to connect to the same IBM i, however the client device is running in Danish CCSID 277. When the user enters their password as PWD@123 and it is passed to the IBM i for verification it will not be valid. When the password was changed while running in CCSID 37, the @ was mapped to x'7C'. When the password is entered while running in the Danish CCSID 277, the @ will map to x'80'. The user will have to enter their password from the Danish CCSID 277 as PWDØ123 to be correct.

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. Setting the password to the default value is not recommended for security reasons.
- 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 or QPWDRULES 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

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.

Add User prompt:
Not shown

CL parameter:
PWDEXP

Length:
4

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 "Password expiration interval" on page 95). 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 53. Possible values for PWDEXP:	
Value	**Description**
*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

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.

Add User prompt:
Not shown

CL parameter:
STATUS

Length:
10

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. "Enabling a user profile" on page 129 shows an example of an adopted authority program to allow a system operator to enable a profile.

The system can disable a profile after a certain number of incorrect password verification 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 54. Possible values for STATUS:

Value	Description
*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

User class is used to control what menu options are shown to the user on IBM i menus. This helps control user access to some system functions.

Add User prompt:
Type of User

CL parameter:
USRCLS

Length:
10

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 55 on page 83 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 55. 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				

Table 55. Default special authorities by user class (continued)					
Special authority	**User classes**				
	***SECOFR**	***SECADM**	***PGMR**	***SYSOPR**	***USER**
*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

The *Assistance level* field in the user profile specifies the default assistance level for the user when the profile is created. The IBM i platform provides three levels of assistance: basic, intermediate, and advanced.

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.

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 56 on page 84 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 56. 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)

Table 56. How assistance levels are stored and changed (continued)		
Action taken	**Version of display shown**	**Version of display stored**
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 "User Options" on page 113.

Table 57. Possible Values for ASTLVL	
Value	**Description**
*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

The *current library* is the library that is specified to be the first user library searched for objects requested by a user. If the user creates objects and specifies *CURLIB, the objects are put in the current library.

Add User prompt:
Default library

CL parameter:
CURLIB

Length:
10

Authority
*USE

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 208 provides more information about using library lists and the current library.

Table 58. Possible values for CURLIB:	
Value	**Description**
*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

You can specify the name of a program to call when a user signs on. Such a program is called an initial program. An initial program runs before the initial menu, if any, is displayed.

Add User prompt:
Sign on program

CL parameter:
INLPGM

Length:
10 (program name) 10 (library name)

Authority:
*USE for program *EXECUTE for library

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 201 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 59. Possible values for INLPGM:

Value	Description
*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 60. Possible values for INLPGM library:

Value	Description
*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

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.

Add User prompt:
First menu

CL parameter:
INLMNU

Length:
10 (menu name) 10 (library name)

Authority
*USE for menu *EXECUTE for library

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 61. Possible values for MENU:

Value	Description
MAIN	The IBM i 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 62. Possible values for MENU library:

Value	Description
*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

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 a tool for preventing users from experimenting on the system.

Add User prompt:
Restrict command line use

CL parameter:
LMTCPB

Length:
10

A user with limited capabilities can only run commands that are defined as being allowed to be used by limited users. The following 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, FTP, REXEC, using the QCAPCMD API, or an option from a command grouping menu. Users are not restricted to perform the following actions:

- Run commands in CL programs that are running a command as a result of taking an option from a menu
- Run 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 63 on page 88 shows the possible values for the Limit capabilities field and what functions are allowed for each value.

Table 63. 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 by default: 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.

Related concepts

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.

Text

The text in the user profile is used to describe the user profile or what it is used for.

Add User prompt:
User description

CL parameter:
TEXT

Length:
50

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 64. Possible values for text:	
Value	**Description**
*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

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.

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.

Table 65. Possible values for SPCAUT:	
Value	**Description**
*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 55 on page 83 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.

*ALLOBJ special authority

All-object (*ALLOBJ) special authority allows the user to access any resource on the system whether 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

The Job control (*JOBCTL) special authority allows a user to 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).

Job control (*JOBCTL) special authority allows the user to perform the following actions:

- 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 211.

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 who abuses *JOBCTL special authority can cause negative effect 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, regardless of whether 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 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. This special authority allows the user to debug a program with only *USE authority to the program and perform the display and alter service functions. It also allows the user to perform trace functions.

The dump function can be performed without *SERVICE authority.

Risks: A user with *SERVICE special authority can display and change confidential information using service functions. 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 the *SERVICE special authority. In this way, only specific users have the ability to perform a trace command, which can 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 IBM i through Application Administration in IBM Navigator for i. The Change Function Usage (CHGFCNUSG) command, 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:

Command	Description
STRCMNTRC	Start Communications Trace
ENDCMNTRC	End Communications Trace
PRTCMNTRC	Print Communications Trace
DLTCMNTRC	Delete Communications Trace
CHKCMNTRC	Check Communications Trace

Command	Description
TRCCNN	Trace Connection (see “Granting access to traces” on page 92)
TRCINT	Trace Internal
STRTRC	Start Job Trace
ENDTRC	End Job Trace
PRTTRC	Print Job Trace
DLTTRC	Delete Job Trace
TRCTCPAPP	Trace TCP/IP Application
WRKTRC	Work with Traces

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

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.

Complete the following steps to limit who can access these trace commands without having *SERVICE authority:

1. In IBM Navigator for i, expand **IBM i Management** > **Users and Groups**.
2. Click **Users** to view a list of user profiles.
3. Right-click the user profile to be altered and select **Application Administration**.
4. In the **Users and Groups Properties** page, in the **Applications** tab, select **Host Applications** from the pull-down menu and click **Go**.
5. Expand **IBM i** > **Service**.
6. Select **Service Trace**.
7. Use the check box to grant or revoke access to trace commands. (To remove the setting for this user use the popup menu next to **Service Trace**.)
8. Click **OK**.

Alternatively, the Change Function Usage (CHGFCNUSG) command can be used to grant users access to the trace commands. Enter CHGFCNUSG FCNID(QIBM_SERVICE_TRACE) USER(user-profile) USAGE(*ALLOWED).

*AUDIT special authority

Audit (*AUDIT) special authority gives the user the ability to view and change auditing characteristics.

A user can perform the following tasks with the *AUDIT special authority:

- Change and display 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.
- Display an object's auditing values.
- Display a user profile's auditing values.
- Run some of the security tool commands, such as PRTADPOBJ.

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.

To prevent general users from viewing auditing information, restrict general users' access to the following information:

- The security audit journal (QAUDJRN)
- Other journals that contain auditing data
- Save files, outfiles, spool files, and printed output that contain auditing information

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. Users with this special authority can add or remove communications configuration information, work with TCP/IP servers, and configure the internet connection server (ICS). Most commands for configuring communications require *IOSYSCFG special authority.

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

In addition, 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

Special environment

The user can operate in the IBM i, 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.

Add User prompt:
Not shown

CL parameter:
SPCENV

Length:
10

Table 66. Possible values for SPCENV:

Value	Description
*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 IBM i 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 IBM i 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 IBM i applications.

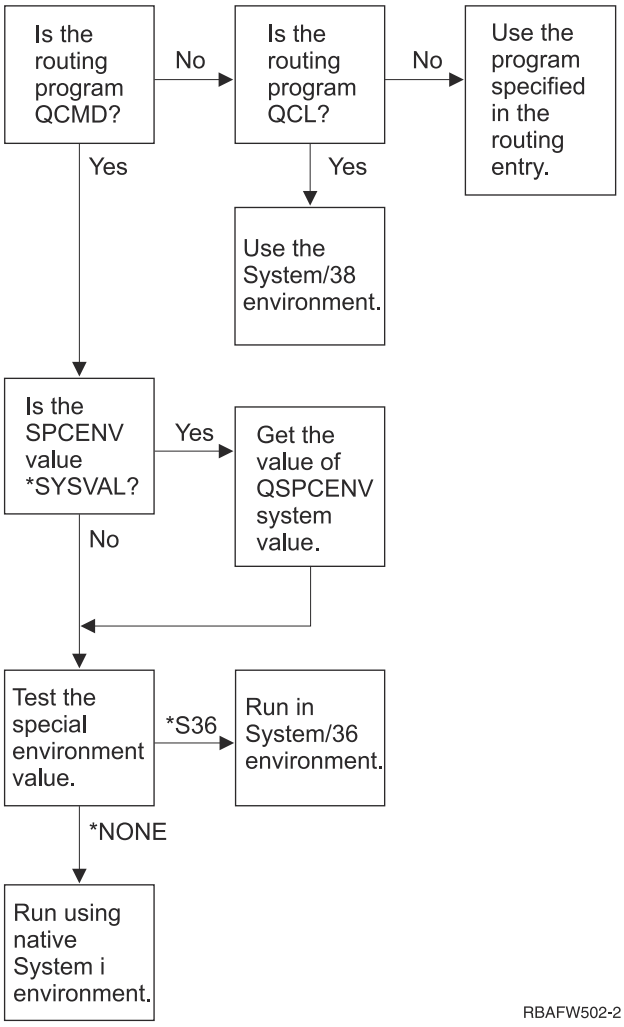

Figure 2. Description of special environment

Description of special environment in Figure 2 on page 94

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 integrated IBM i environment.

Display sign-on information

The Sign-on Information display is a tool for users to monitor their profiles and to detect attempted misuse. The Display sign-on information field specifies whether the Sign-on Information display is shown when the user signs on.

Add User prompt:
Not shown

CL parameter:
DSPSGNINF

Length:
7

Figure 3 on page 95 shows the display. Password expiration information is only shown if the password expires within the password expiration warning days.

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

Password verifications not valid . . . . . :   3

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

Figure 3. Sign-On Information Display

Table 67. Possible values for DSPSGNINF:	
Value	**Description**
*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: 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

The password expiration interval controls the number of days that a valid password can be used before it must be changed.

Add User prompt:
Not shown

CL parameter:
PWDEXPITV

Length:
5,0

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. "Password expiration interval" on page 95 shows an example of the password expiration message.

Table 68. Possible values for PWDEXPITV:	
Value	**Description**
*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 to require users with *SERVICE,

*SAVSYS, *SECADM, or *ALLOBJ special authorities to change passwords more frequently than other users.

Block Password Change

The block password change parameter specifies the time period during which a password is blocked from being changed after the prior successful password change operation.

Add User prompt:
Not shown

CL parameter:
PWDCHGBLK

Length:
10

This parameter value does not restrict password changes made by the Change User Profile (CHGUSRPRF) command. In addition, this parameter value is not enforced if the set password to expired (PWDEXP) field in the user profile has a value of *YES. This enables a security administrator to create a user profile with an expired password and still permit the user to sign-on and change the password (once) without being restricted by the block password change system value.

Table 69. Possible values for PWDCHGBLK:

Value	Description
*SYSVAL	The QPWDCHGBLK system value is used.
*NONE	The password can be changed at any time.
1 - 99	A password cannot be changed within the specified number of hours after the prior successful password changed operation.

Recommendation: Set the parameter to *SYSVAL unless you notice unusual password change activity for a specific user. In this case, you can use a value, such as 2, to limit the user's password change frequency.

Local password management

The Local password management (LCLPWDMGT) parameter controls whether the user profile password is managed locally. When the password is not management locally, users cannot access the system by direct sign-on, but through other platforms.

If the password is managed locally, then the password is stored locally with the user profile. This is the traditional method of storing the password.

Add User prompt:
Not shown

CL parameter:
LCLPWDMGT

Length:
10

If the password is not being managed locally, then the local IBM i password is set to *NONE. The password value specified in the password parameter will be sent to other IBM products that do password synchronization, such as IBM i Integration for Windows Server. Users will not be able to change their passwords using the Change Password (CHGPWD) command. In addition, users will not be able to sign on to the system directly. Specifying this value will affect other IBM products that do password synchronization, such as IBM i Integration for Windows Server.

This parameter should not be set to *NO unless the user only needs to access the system through some other platform, such as Windows Server.

Table 70. Possible values for LCLPWDMGT:	
Value	**Description**
*YES	The password is managed locally.
*NO	The password is not managed locally.

Limit device sessions

The Limit device sessions field controls whether the number of device sessions allowed for a user is limited. The value does not restrict the use of the System Request menu or a second sign-on from the same device.

Add User prompt:
Not shown

CL parameter:
LMTDEVSSN

Length:
7

Table 71. Possible values for LMTDEVSSN:	
Value	**Description**
*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.
0	The user is not limited to a specific number of device sessions. This value has the same meaning as *NO.
1	The user is limited to a single device session. This value has the same meaning as *YES.
2 - 9	The user is limited to the specified number of device sessions.

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

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.

Add User prompt:
Not shown

CL parameter:
KBDBUF

Length:
10

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 72. Possible values for KBDBUF:

Value	Description
*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

You can specify the maximum amount of auxiliary storage that the system uses to store permanent objects that a user profile owns. This includes objects that the system places in the temporary library (QTEMP) during a job.

Add User prompt:
Not shown

CL parameter:
MAXSTG, MAXSTGLRG

Length:
11,0 (MAXSTG), 20 (MAXSTGLRG)

The MAXSTGLRG parameter allows a larger maximum storage value than the MAXSTG parameter.

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 size of auxiliary storage:

- 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) or MAXSTGLRG(*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) or MAXSTGLRG(*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 system assigns storage for the descriptions of objects that are placed in a library to the owner of that library. This is true even if the objects are owned by another user profile. Examples of such descriptions are text and program references.
- The system assigns storage to the user profile for temporary objects that are used during job processing. Examples of such objects are commitment control blocks, file editing spaces, and documents.

Table 73. Possible values for MAXSTG and MAXSTGLRG:

Value	Description
*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

The priority limit in the user profile determines the maximum scheduling priorities (job priority and output priority) that are allowed for any jobs the user submits. Priority limit controls the job's priority when it is submitted. It also controls any changes made to the job's priority while the job is waiting in the queue, or when the job runs.

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 in the job queue. You can set the job's priority in the job description or by using the submit command.

Output priority:
Determines the scheduling priority for any output created by the job on the output queue. You can set the output priority in the job description or when you use the submit command.

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 74. Possible values for PTYLMT:

Value	Description
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.

Table 74. Possible values for PTYLMT: (continued)	
Value	**Description**
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 often 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

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.

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.

See the Work management topic for more information about job descriptions and their uses.

Table 75. Possible values for JOBD:	
Value	**Description**
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 76. Possible values for JOBD Library:	
Value	**Description**
*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 (user library list) system value.

Group profile

The group profile (GRPPRF) parameter specifies if the user is a member of a 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.

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.

Note:
Adopted authority is not used to check for *OBJMGT authority to the group profile. For more information about adopted authority, see "Objects that adopt the owner's authority" on page 153.

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.

When the GRPPRF value is changed, the change takes effect the next time the user signs on or the next time a job swaps to the user profile using a profile handle or profile token, which was obtained after the change occurred.

See "Planning group profiles" on page 240 for more information about using group profiles.

Table 77. Possible values for GRPPRF

Value	Description
*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

If the user is a member of a group, you can 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 a value other than *NONE for the Group profile field.

Add User prompt:
Not shown

CL parameter:
OWNER

Length:
10

When the Owner value is changed, the change takes effect the next time the user signs on or the next time a job swaps to the user profile using a profile handle or profile token obtained after the change has occurred.

Table 78. Possible values for Owner:	
Value	**Description**
*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. Using *GRPPRF is a security risk as all members of the group obtain all authority and ownership rights to objects created by this user profile 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

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.

Add User prompt:
Not shown

CL parameter:
GRPAUT

Length:
10

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.

When the GRPAUT value is changed, the change takes effect the next time the user signs on or the next time a job swaps to the user profile using a profile handle or profile token obtained after the change has occurred.

Note: Using the GRPAUT parameter with a value other than *NONE gives all other users who are members of the group profile that is specified on the GRPPRF parameter authority to objects created by this user.

Table 79. Possible values for GRPAUT:	
Value	**Description**
*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.

Related reference

Defining how information can be accessed
You can define what operations can be preformed on objects, data, and fields.

Group authority type

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.

Add User prompt:
Not shown

CL parameter:
GRPAUTTYP

Length:
10

The GRPAUTTYP parameter works with the OWNER, GRPPRF, and GRPAUT parameters to determine the group's authority to a new object.

When the GRPAUTTYP value is changed, the change takes effect the next time the user signs on or the next time a job swaps to the user profile using a profile handle or profile token obtained after the change has occurred.

Table 80. Possible values for GRPAUTTYP: [1]

Value	Description
*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. This value can be specified only when GRPAUT is not *NONE.

1 Private authority and primary group authority provide the same access to the object for members of the group, but they might have different performance characteristics. "Primary group for an object" on page 148 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 that must be accessed by members of the group profile.

Supplemental groups

You can specify supplemental groups when creating or changing a user profile. The user cannot have supplemental group profiles if the GRPPRF parameter is *NONE.

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 153.

You can 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.

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 marks it as a group profile. The system also generates a group identification number (gid) for the group profile, if it does not already have one.

When the SUPGRPPRF value is changed, the change takes effect the next time the user signs on or the next time a job swaps to the user profile using a profile handle or profile token obtained after the change has occurred.

See "Planning group profiles" on page 240 for more information about using group profiles.

Table 81. Possible values for SUPGRPPRF

Value	Description
*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. The profile name specified for GRPPRF can also be specified as one of the 15 supplemental group profiles.

Accounting code

Specifying the accounting code allows you to gather information about the system resources used by a job.

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.

When the *accounting code* value is changed, the change takes effect the next time the user signs on or the next time a job, which runs using the user profile's accounting code value, is started.

See the Work management topic for more information about job accounting.

Table 82. Possible values for ACGCDE:

Value	Description
*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

A document password controls the accessibility and distribution of personal mail when viewed by people who are working on behalf of the user. The document password is supported by some Document Interchange Architecture (DIA) products, such as the Displaywriter.

Add User prompt:
Not shown

CL parameter:
DOCPWD

Table 83. Possible values for DOCPWD:	
Value	**Description**
*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

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.

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.

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 84. Possible values for MSGQ:	
Value	**Description**
*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 85. Possible values for MSGQ Library:	
Value	**Description**
*LIBL	The library list is used to locate the message queue. If the message queue does not exist, you cannot specify *LIBL.

Table 85. Possible values for MSGQ Library: (continued)	
Value	**Description**
*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: Give each user profile a unique message queue, preferably with the same name as the user profile.

Delivery

The delivery mode of a message queue determines whether the user is interrupted when a new message arrives on the queue.

Add User prompt:
Not shown

CL parameter:
DLVRY

Length:
10

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 86. Possible values for DLVRY:	
Value	**Description**
*NOTIFY	The job to which the message queue is assigned is notified when a message arrives at the message queue. For interactive jobs at a workstation, the audible alarm sounds and the message-waiting light turns 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 sounds (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

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.

Add User prompt:
Not shown

CL parameter:
SEV

Length:
2,0

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 87. Possible values for SEV:

Value	Description
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

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).

Add User prompt:
Default printer

CL parameter:
PRTDEV

Length:
10

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 Basic printing topic.

Table 88. Possible values for PRTDEV:

Value	Description
*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

Both interactive and batch processing can 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.

Add User prompt:
Not shown

CL parameter:
OUTQ

Length:
10 (output queue name) 10 (library name)

Authority:
*USE for output queue *EXECUTE for library

An output queue does not need 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 Advanced Function Presentation topic.

Table 89. Possible values for OUTQ:	
Value	**Description**
*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 90. Possible values for OUTQ library:	
Value	**Description**
*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

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

Add User prompt:
Not shown

CL parameter:
ATNPGM

Length:
10 (program name) 10 (library name)

Authority:
*USE for program

*EXECUTE for library

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 201 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 91. Possible values for ATNPGM:	
Value	**Description**
*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 92. Possible values for ATNPGM Library:	
Value	**Description**
*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

Sort sequence is used for this user's output. You can use system-provided sort tables or create your own. A sort table can be associated with a particular language identifier on the system.

Add User prompt:
Not shown

CL parameter:
SRTSEQ

Length:
10 (value or table name) 10 (library name)

Authority:
*USE for table *EXECUTE for library

Table 93. Possible values for SRTSEQ:	
Value	**Description**
*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 94. Possible values for SRTSEQ Library:	
Value	**Description**
*LIBL	The library list is used to locate the table specified for the SRTSEQ value.

Table 94. Possible values for SRTSEQ Library: (continued)	
Value	**Description**
*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

You can specify the language identifier to be used by the system for the user.

Add User prompt:
Not shown

CL parameter:
LANGID

Length:
10

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 95. Possible values for LANGID:	
Value	**Description**
*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

You can specify the country or region identifier to be used by the system for the user.

Add User prompt:
Not shown

CL parameter:
CNTRYID

Length:
10

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 96. Possible values for CNTRYID:	
Value	**Description**
*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

You can specify the coded character set identifier to be used by the system for the user.

Add User prompt:
Not shown

CL parameter:
CCSID

Length:
5,0

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 97. Possible values for CCSID:

Value	Description
*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

The *CHRIDCTL* attribute controls the type of coded character set conversion that occurs for display files, printer files and panel groups.

Add User prompt:
Not shown

CL parameter:
CHRIDCTL

Length:
10

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 98. Possible values for CHRIDCTL:

Value	Description
*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 is necessary. 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

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

Add User prompt:
Not shown

CL parameter:
SETJOBATR

Length:
160

Table 99. Possible values for SETJOBATR:	
Value	**Description**
*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.
*CCSID	The coded character set identifier (CCSID) 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.

Any combination of the following values can be specified:

- *CCSID
- *DATFMT
- *DATSEP
- *DECFMT
- *SRTSEQ
- *TIMSEP

Locale

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

Add User prompt:
Not shown

CL parameter:
LOCALE

Table 100. Possible values for LOCALE:	
Value	**Description**
*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

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.

Add User prompt:
Not shown

CL parameter:
USROPT

Length:
240 (10 characters each)

Table 101. Possible values for USROPT:

Value	Description
*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 158 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 typically 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

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.

Add User prompt:
Not shown

CL parameter:
UID

Length:
10,0

Table 102. Possible values for UID:

Value	Description
*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

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 must have a gid.

Add User prompt:
Not shown

CL parameter:
GID

Length:
10,0

Table 103. Possible values for GID:

Value	Description
*NONE	This profile does not have a gid. This value must be specified if the user profile is a member of a group (GRPPRF is not *NONE).
*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 might 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

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.

Add User prompt:
Not shown

CL parameter:
HOMEDIR

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 104. Possible values for HOMEDIR:

Value	Description
*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.

EIM association

The EIM association specifies whether an Enterprise Identity Mapping (EIM) association should be added to an EIM identifier for this user. Optionally, the EIM identifier can also be created if it does not already exist.

Add User prompt:
Not shown

CL parameter:
EIMASSOC

Notes:

1. The EIM association information is not stored in the user profile. This information is not saved or restored with the user profile.
2. If this system is not configured for EIM, then no processing is done. Not being able to perform EIM operations does not cause the command to fail.

Table 105. Possible values for EIMASSOC, single values:	
Single values	
*NOCHG	EIM association will not be added.

Table 106. Possible values for EIMASSOC, element 1:	
Element 1: EIM identifier **Specifies the EIM identifier for this association.**	
*USRPRF	The name of the EIM identifier is the same name as the user profile.
character-value	Specifies the name of the EIM identifier.

Table 107. Possible values for EIMASSOC, element 2:	
Element 2: Association type **Specifies the type of association. It is recommended that a target association is added for an IBM i user.** **Target associations are primarily used to secure existing data. They are found as the result of a mapping lookup operation (for example, eimGetTargetFromSource()), but cannot be used as the source identity for a mapping lookup operation.** **Source associations are primarily used for authentication purposes. They can be used as the source identity of a mapping lookup operation, but will not be found as the target of a mapping lookup operation.** **Administrative associations are used to show that an identity is associated with an EIM identifier, but cannot be used as the source for, and will not be found as the target of, a mapping lookup operation.**	
*TARGET	Process a target association.
*SOURCE	Process a source association.
*TGTSRC	Process both a target and a source association.
*ADMIN	Process an administrative association.
*ALL	Process all association types.

Table 108. Possible values for EIMASSOC, element 3:	
Element 3: Association action	
*REPLACE	Associations of the specified type will be removed from all EIM identifiers that have an association for this user profile and local EIM registry. A new association will be added to the specified EIM identifier.
*ADD	Add an association.
*REMOVE	Remove an association.

Table 109. Possible values for EIMASSOC, element 4:	
Element 4: Create EIM identifier **Specifies whether the EIM identifier should be created if it does not already exist.**	
*NOCRTEIMID	EIM identifier does not get created.
*CRTEIMID	EIM identifier gets created if it does not exist.

User expiration date

The User expiration date can be used to specify the date at which the user profile is automatically disabled.

Add User prompt:
Not shown

CL parameter:
USREXPDATE

Length:
6

The User expiration date field allows a security administrator to indicate that the user profile will expire on a specific date. If User expiration interval is used, this date is calculated by the system.

Table 110. Possible values for USREXPDATE:	
Value	**Description**
*NONE	The user profile does not have an expiration date.
*USREXPITV	The user expiration date is to be calculated using the value specified in the User expiration interval (USREXPITV) parameter.
user-expiration-date	Specifies the date when the user profile expires. The date must be specified in the job date format.

User expiration interval

The User expiration interval controls the number of days before the user profile is automatically disabled.

Add User prompt:
Not shown

CL parameter:
USREXPITV

Length:
5,0

The User expiration interval field allows a security administrator to indicate in the user profile the number of days before the user profile will expire and be automatically disabled. If a value is specified for User

expiration interval when a user profile is created or when an expired user profile is re-enabled, the User expiration date is generated by the system using the expiration interval.

Table 111. Possible values for USREXPITV:	
Value	**Description**
user-expiration-interval	Specify a number from 1 through 366.

Authority

The Authority field specifies the public authority to the user profile.

Add User prompt:
Not shown

CL parameter:
AUT

The authority to a profile controls many functions associated with the profile, such as:

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

Table 112. Possible values for AUT:	
Value	**Description**
*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 136 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

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.

Add User prompt:
Not shown

CL parameter:
OBJAUD

Length:
10

Object auditing for a user profile cannot be specified on any user profile commands. Use the CHGUSRAUD command to specify object auditing for a user. Only a user with *AUDIT special authority can use the CHGUSRAUD command.

Table 113. Possible values for OBJAUD:

Value	Description
*NONE	The OBJAUD value for objects determines whether object auditing is done for this user.
*ALL	If the OBJAUD value for an object specifies *USRPRF, an audit record is written when this user changes or reads the object.
*CHANGE	If the OBJAUD value for an object specifies *USRPRF, an audit record is written when this user changes the object.
*NOTAVL	This value is displayed to indicate that the parameter value is not available to the user because the user does not have either *AUDIT or *ALLOBJ special authority. The parameter value cannot be set to this value.

Table 114 on page 118 shows how the OBJAUD values for the user and the object work together:

Table 114. Auditing performed for object access

OBJAUD value for object	OBJAUD value for user		
	*NONE	*CHANGE	*ALL
*ALL	Change and Use	Change and Use	Change and Use
*CHANGE	Change	Change	Change
*NONE	None	None	None
*USRPRF	None	Change	Change and Use

Related tasks

Planning the auditing of object access

The IBM i operating system provides the ability to log accesses to an object in the security audit journal by using system values and the object auditing values for users and objects. This is called *object auditing*.

Action auditing

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 and QAUDLVL2 system values.

Add User prompt:
Not shown

CL parameter:
AUDLVL

Length:
640

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.

Note: Consider using the CHGUSRAUD command to set action auditing on your security officer and other highly privileged users. Auditing the actions of the security officers and other privileged users is recommended as these users will be authorized to perform many or all system functions. They also have access to highly sensitive data objects on the server.

Table 115. Possible values for AUDLVL:	
Value	**Description**
*NONE	The QAUDLVL system value controls action auditing for this user. No additional auditing is done.
*NOTAVL	This value is displayed to indicate that the parameter value is not available to the user because the user does not have either *AUDIT or *ALLOBJ special authority. The parameter value cannot be set to this value.
*AUTFAIL	Authorization failures are audited.
*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.
*JOBBAS	Job base functions are audited.
*JOBCHGUSR	Changes to a thread's active user profile or its group profiles are audited.
*JOBDTA[1]	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.
*NETBAS	Network base functions are audited.
*NETCLU	Cluster or cluster resource group operations are audited.
*NETCMN [3]	Networking and communications functions are audited.
*NETFAIL	Network failures are audited.
*NETSCK	Sockets tasks are audited.
*NETSECURE	Secure network connections are audited.
*NETUDP	User Datagram Protocol (UDP) traffic is audited.
*OPTICAL	All optical functions are audited.
*PGMADP	Obtaining authority to an object through a program that adopts authority is logged.
*PGMFAIL	Program failures are audited.
*PRTDTA	Printing functions with parameter SPOOL(*NO) are audited.
*SAVRST	Save and restore operations are logged.
*SECCFG	Security configuration is audited.
*SECDIRSRV	Changes or updates when doing directory service functions are audited.
*SECIPC	Changes to interprocess communications are audited.
*SECNAS	Network authentication service actions are audited.
*SECRUN	Security run time functions are audited.
*SECSCKD	Socket descriptors are audited.

Table 115. Possible values for AUDLVL: (continued)	
Value	**Description**
*SECURITY[2]	Security-related functions are logged.
*SECVFY	Use of verification functions are audited.
*SECVLDL	Changes to validation list objects are audited.
*SERVICE	Using service tools is logged.
*SPLFDTA	Actions performed on spooled files are logged.
*SYSMGT	Use of systems management functions is logged.

1

*JOBDTA includes two values that are *JOBBAS and *JOBCHGUSR, which enable you to better customize your auditing. If both of the values are specified, you will get the same auditing as if just *JOBDTA is specified.

2

*SECURITY is composed of several values to enable you to better customize your auditing. If all of the values are specified, you will get the same auditing as if just *SECURITY is specified. These values are as follows.

- *SECCFG
- *SECDIRSRV
- *SECIPC
- *SECNAS
- *SECRUN
- *SECSCKD
- *SECVFY
- *SECVLDL

3

*NETCMN is composed of several values to enable you to better customize your auditing. The following values make up *NETCMN:

- *NETBAS
- *NETCLU
- *NETFAIL
- The Mail and DHCP functions from *NETSCK

Related reference

Planning the auditing of actions

The QAUDCTL (audit control) system value, the QAUDLVL (audit level) system value, the QAUDLVL2 (audit level extension) system value, and the AUDLVL (action auditing) parameter in user profiles work together to control action auditing.

Additional information associated with a user profile

This topic discusses the private authorities, owned object information, and primary group object information that are associated with a user profile.

Related reference

How security information is stored

Planning adequate backup and recovery procedures for security information requires understanding how the information is stored and saved.

Private authorities

All of the private authorities that a user has to objects are stored with the user profile. When a user needs authority to an object, the user's private authorities might be searched.

"Flowchart 3: How user authority to an object is checked" on page 177 provides more information about authority checking.

You can display a user's private authorities to library-based objects by using the Display User Profile command:

```
DSPUSRPRF user-profile-name TYPE(*OBJAUT)
```

You can work with a user's private authorities to library- and directory-based objects using the Work with Objects by Private Authority (WRKOBJPVT) command. To change a user's private authorities, you can use the commands that work with object authorities, such as Edit Object Authority (EDTOBJAUT).

You can copy all of the private authorities from one user profile to another using the Grant User Authority (GRTUSRAUT) command. See "Copying authority from a user" on page 168 for more information.

Primary group authorities

The names of all of the objects for which the profile is the primary group are stored with the group profile.

You can display the library-based 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

Because the size of a user profile can affect your performance, it is suggested that you do not assign all (or nearly all) objects to only one owning profile.

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.

Digital ID authentication

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

The digital ID APIs create, distribute, and manage digital certificates associated with user profiles. See Digital certificate management APIs 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 topic describes the commands and displays you use to create, change, and delete user profiles on the IBM i operating system.

You must have *SECADM special authority to create, change, or delete user profiles.

Creating user profiles

You can create a user profile by 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 or using IBM Navigator for i .

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: You cannot use the Create User Profile(CRTUSRPRF) command to create a user profile 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 Work with User Profiles (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       Name
User password  . . . . . .   *NONE         Character value, *USRPRF...
Set password to expired  . . . .   *YES          *NO, *YES
Status . . . . . .   *ENABLED      *ENABLED, *DISABLED
User class . . . . . .   *USER         *USER, *SYSOPR, *PGMR...
Assistance level . . . . . .   *SYSVAL       *SYSVAL, *BASIC, *INTERMED...
Current library  . . . . . .   *CRTDFT       Name, *CRTDFT
Initial program to call  . . . .   *NONE         Name, *NONE
  Library  . . . . . .                     Name, *LIBL, *CURLIB
Initial menu . . . . . .   MAIN          Name, *SIGNOFF
  Library  . . . . . .   QSYS          Name, *LIBL, *CURLIB
Limit capabilities . . . . . .   *NO           *NO, *PARTIAL, *YES
Text 'description' . . . . . .   *BLANK
```

The Create User Profile display shows all of 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 (Create User Profile) 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

You can use the Work with User Enrollment option to add users to the system.

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        Name
User description  . . . .
Password  . . . . . .   NEWUSER
Type of user  . . . . . .   *USER          Type, F4 for list
User group  . . . . . .   *NONE          Name, F4 for list

Restrict command line use   N              Y=Yes, N=No

Default library . . . . .                  Name
Default printer . . . . .   *WRKSTN        Name, *WRKSTN, F4 for list
Sign on program . . . . .   *NONE          Name, *NONE
  Library . . . . . .                      Name

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

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 . . . . . .
```

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.

Copying user profiles

You can create a user profile by copying another user profile or a group profile.

You might 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.

Related concepts

Group profiles

A *group profile* is a special type of user profile. Rather than giving authority to each user individually, you can use a group profile to define authority for a group of users.

Copying from the Work with User Profiles display

You can copy the information of a user profile 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 of the values from the copy-from user profile are shown on the Create User Profile display, except the following fields:

User profile
Blank. Must be filled in.

Password
CRTUSRPRF command default

Document password
*NONE

Message queue
*USRPRF

Locale job attributes
*SYSVAL

Locale
*SYSVAL

User Identification Number
*GEN

Group Identification Number
*NONE

Home directory
*USRPRF

EIM Association
*NOCHG

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 are not copied.

Copying from the Work with User Enrollment display

You can also copy user profiles 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

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

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

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

EIM Association
*NOCHG

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 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.

Related concepts
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.

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. Before you can delete such user profiles, you must delete any objects owned by the profile or transfer ownership of those objects to another 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 WRKOBJPGP command 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 or SUPGRPPRF field in each member profile before deleting the group profile.

Using the Delete User Profile command

To delete a user profile, 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
- EIM associations

```
                       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  .
EIM association  . . . . . . . .   *DLT          *DLT, *NODLT
```

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

You can use the Remove User option on the Work with User Enrollment display to delete a user profile.

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 Private Authorities

You can use the Work with Objects by Private Authorities (WRKOBJPVT) command to display and work with objects for which a profile has private authority.

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 password verification attempts, you might need to enable the profile by changing the profile status to *ENABLED.

To enable a user profile, you must have *SECADM special authority, *OBJMGT authority, 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, *OBJMGT authority, 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

You can 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  Level      Level                   Local
User      Group     Last      0 or 1     2 or 3      Netserver   Pwd
Profile   Profile   Changed   Password   Password    Password    Mgt
ANGELA              04/21/0x    *YES       *NO        *YES       *YES
ARTHUR              07/07/0x    *YES       *YES       *YES       *YES
CAROL1              05/15/0x    *YES       *YES       *YES       *YES
CAROL2              05/15/0x    *NO        *NO        *NO        *NO
CHUCKE              05/18/0x    *YES       *NO        *YES       *YES
DENNISS             04/20/0x    *YES       *NO        *YES       *YES
DPORTER             03/30/0x    *YES       *NO        *YES       *YES
GARRY               08/04/0x    *YES       *YES       *YES       *YES
JANNY               03/16/0x    *YES       *NO        *YES       *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.
- 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. “Analyzing user profiles” on page 310 gives suggestions for reports.

Types of user profile reports

You can generate user profile reports by using the Print User Profile (PRTUSRPRF) command or the Analyze Default Password (ANZDFTPWD) command.

- Print User Profile (PRTUSRPRF)

  This command generates reports that contain information about the user profiles on the system. Four different variations of this report 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 generates a report about all of the user profiles on the system that have a default password and allows you 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 integrated file system cannot be changed.
- You might 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 IBM i user profile to rename a user does not rename any application profiles the user might 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, while 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

You can use the Change User Auditing (CHGUSRAUD) command to set the audit characteristics for users.

To use this command, you must have *AUDIT special authority.

```
          Change User Audit (CHGUSRAUD)

Type choices, press Enter.

User profile . . . . . . . . . .   HOGANR
               + for more values   JONESS
Object auditing value  . . . . .   *SAME
User action auditing . . . . . .   *CMD
               + for more values   *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 that you specify are not added to the current AUDLVL values for the users but rather they replace the current AUDLVL values.

If you have either *ALLOBJ or *AUDIT special authority, you can use the Display User Profile (DSPUSRPRF) command to see audit characteristics for a user.

Working with profiles in CL programs

You can work with user profiles within a CL program.

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 section 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 65 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 need 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

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 operations:

- 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.

Note: All adopted authority will be suppressed before the exit programs are called. This means that the exit program may not have authority to access the user profile object.

Related information
Exit programs

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.

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

Note: All IBM-supplied user profiles except for QSECOFR are shipped with a password of *NONE and are not intended for sign-on. These profiles are used by the IBM i operating system. Therefore, signing on with these profiles or using the profiles to own user (non-IBM supplied) objects is not recommended.

Related concepts
IBM-supplied user profiles
You can perform auditing tasks on IBM-supplied user profiles by verifying their passwords.

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

You can manage service tools user IDs using system service tools (SST), dedicated service tools (DST), and CL commands.

See Managing service tools user IDs for more information on creating, changing, and deleting service tools user IDs.

Related concepts

IBM-supplied user profiles

You can perform auditing tasks on IBM-supplied user profiles by verifying their passwords.

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

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

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

Chapter 7, “Designing security,” on page 221 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 172 provides detailed flowcharts and notes about how the system checks authority. You might find it useful to consult this information as you read the explanations that follow.

Related concepts

Resource security
The ability to access an object is called *authority*. Resource security on the IBM i operating system enables you to control object authorities by defining who can use which objects and how those objects can be used.

Overall recommendations for security design
Keeping your security design as simple as possible makes it easier to manage and audit security. It also improves application performance and backup performance.

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 authority** 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 can 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 can be given authority to use objects on the system. This is one type of private authority.

Group authority:
Group profiles can 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 can provide better performance than private

authority granted to a group profile. Only a user profile with a group identification number (gid) can be the primary group for an object. Primary group authority is not considered private authority.

Defining how information can be accessed

You can define what operations can be preformed on objects, data, and fields.

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.
3. **Field authority** defines what operations can be performed on the data fields.

Table 116 on page 136 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, “Authority required for objects used by commands,” on page 371 provides information about the authority that is required to perform a specific function.

Table 116. 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].
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.

Table 116. Description of authority types (continued)		
Authority	**Name**	**Functions allowed**
*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 143 for more information.

Related tasks

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 to update special authorities the next time you perform an initial program load (IPL).

Related reference

Group authority
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.

Commonly used authorities

You can specify certain sets of objects and data 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 117 on page 137 shows the system-defined authorities available using the object authority commands and displays.

Table 117. System-defined authority				
Authority	***ALL**	***CHANGE**	***USE**	***EXCLUDE**
Object Authorities				

Table 117. System-defined authority (continued)				
Authority	***ALL**	***CHANGE**	***USE**	***EXCLUDE**
*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 118 on page 138 shows additional system-defined authorities that are available using the WRKAUT and CHGAUT commands:

Table 118. 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 119 on page 138 shows how the LAN Server permissions map to object and data authorities:

Table 119. LAN server permissions	
Authority	**LAN server permissions**
*EXCLUDE	None
Object Authorities	

Table 119. LAN server permissions (continued)	
Authority	**LAN server permissions**
*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

You can use library security to protect information.

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, “Authority required for objects used by commands,” on page 371 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 can do the following actions:

- 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 might 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.

Related concepts

Planning libraries

A library is like a directory used to locate the objects in the library. Many factors affect how you choose to group your application information into libraries and manage libraries.

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 to an object is requested 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 can potentially run a different version of a program.

Related reference

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.

Field authorities

You can specify field authorities for database files.

Field authorities are supported for database files. Authorities supported are Management, Alter, Reference, Read, Add, 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
*PUBLIC                *CHANGE     X     X     X       X       X
PGMR1                  *ALL        X     X     X       X       X
USER1                  *USE        X                           X
USER2                  USER DEF    X           X               X
USER3                  USER DEF    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 List prompt will be displayed. If F16 is pressed, the previous position to operation will be repeated.

Field authorities include the following options:

- The Print Private Authority (PRTPVTAUT) command has a field that indicates when a file has field authorities.
- The Display Object Authority (DSPOBJAUT) command has an 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 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

This section provides information about security in 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 might 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.

Related information

System/38 Environment Programming

Recommendation for System/38 Environment

This topic includes a list of recommendations for the 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

You can use directory security to protect information.

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 might 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 authorities that the users have for 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 in 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 that 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 will be lost. You can restore the association for these objects if you have saved security data from a previous release of IBM i 7.3 or higher. Run RSTUSRPRF USRPRF(*NEW) and then RSTAUT to restore these associations.

See the topic “Advantages of using an authorization list” on page 169 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 that 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 can use authorization lists to secure IBM-supplied objects. For example, you might 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. You can restore the links for objects in QSYS if you have saved security data from IBM i 7.3 or higher. Run RSTUSRPRF USRPRF(*NEW) and then RSTAUT to restore these links.

Authority for new objects in a library

You can specify the 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 for most commands, the public authority for the object is set to the CRTAUT value for the library.

For example, assume that 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 that 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 149 shows more examples of how the system assigns ownership and authority to new objects.

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

You need to consider the risks when you change the Create Authority (CRTAUT) for an application library.

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 might allow unauthorized access to new objects created in the library.

Authority for new objects in a directory

You can specify the authority for new objects in a directory.

When you create a new directory using the CRTDIR (Make Directory), MD (Make Directory) or MKDIR (Make Directory) commands, you specify the data authority and object authority that the public receives for the new directory. If you use the default *INDIR option, the authority for the created directory is determined from its parent directory. Otherwise, you can specify the specific required authority.

When you create a new directory using the mkdir()--Make Directory API, the owner, primary group, and public object authorities for the created directory are determined from the directory in which it is being created in while the owner, primary group, and public data authorities are determined by the mode that is specified on the API call.

The following two examples show different results when you create a new directory with various options.

The first example creates a new directory in the "root"(/) file system using the CRTDIR command and specify *PUBLIC authority.

Starting conditions: Authorities on parent directory:

```
                                    Display Authority

Object . . . . . . . . . . . . . :   /sanders/mytest
Owner  . . . . . . . . . . . . . :   SANDERS
Primary group  . . . . . . . . . :   SANDERSGP3
Authorization list . . . . . . . :   *NONE

              Data      -----Object Authorities-----
User         Authority  Exist    Mgt    Alter    Ref
*PUBLIC      *RWX        X        X      X        X
SANDERS      *RW
SANDERSGP3   *RX
QPGMR        *RWX
QTCM         *RWX        X        X      X        X
```

User SANDERS issues the following command:
CRTDIR DIR('/sanders/mytest/deletemepub') DTAAUT(*R) OBJAUT(*NONE)

Results: Authorities on created directory:

```
                                    Display Authority

Object . . . . . . . . . . . . . :   /sanders/mytest/deletemepub
Owner  . . . . . . . . . . . . . :   SANDERS
Primary group  . . . . . . . . . :   SANDERSGP3
Authorization list . . . . . . . :   *NONE

              Data      -----Object Authorities-----
User         Authority  Exist    Mgt    Alter    Ref
*PUBLIC      *R
SANDERS      *RWX
SANDERSGP3   *RX
```

Notes:

1. The *PUBLIC data and object authorities are set based on the DTAAUT and OBJAUT parameters.
2. The owner's (SANDERS) data authorities are set to *RWX but the object authorities are inherited from the parent directory's owner. This means that the owner of this directory has no object authorities to the new directory because the owner of the parent directory has no object authorities to the parent directory.
3. The new directory has a primary group profile of SANDERSGP3 because the parent directory has SANDERSGP3 as its primary group profile.

The second example shows how all authorities are inherited from the parent directory when you create a new directory in the "root" (/) file system using the CRTDIR command .

Starting conditions: Authorities on parent directory:

```
                                   Display Authority

Object . . . . . . . . . . . . . :   /sanders/mytest
Owner  . . . . . . . . . . . . . :   SANDERS
Primary group  . . . . . . . . . :   SANDERSGP3
Authorization list . . . . . . . :   *NONE

              Data      -----Object Authorities-----
User         Authority  Exist    Mgt    Alter    Ref
*PUBLIC      *RWX        X        X      X        X
SANDERS      *RW
SANDERSGP3   *RX
QPGMR        *RWX
QTCM         *RWX        X        X      X        X
```

User SANDERSUSR issues the following command:
CRTDIR DIR('/sanders/mytest/deletemepub')

Results: Authorities on created directory:

```
                                   Display Authority

Object . . . . . . . . . . . . . :   /sanders/mytest/deletemepub
Owner  . . . . . . . . . . . . . :   SANDERSUSR
Primary group  . . . . . . . . . :   SANDERSGP3
Authorization list . . . . . . . :   *NONE

              Data      -----Object Authorities-----
User         Authority  Exist    Mgt    Alter    Ref
*PUBLIC      *RWX        X        X      X        X
SANDERSUSR   *RWX
SANDERSGP3   *RX
QPGMR        *RWX
QTCM         *RWX        X        X      X        X
SANDERS      *RW
```

Notes:

1. The *PUBLIC data and object authorities are inherited from the parent directory; therefore, the data authority is set to *RWX with all object authorities.
2. The owner's (SANDERSUSR) data authorities are set to *RWX but the object authorities are inherited from the parent directory's owner. This means that the owner of this directory has no object authorities to the new directory because the owner of the parent directory has no object authorities to the parent directory.
3. The new directory has a primary group profile of SANDERSGP3 because the parent directory has SANDERSGP3 as its primary group profile.
4. All users who are privately authorized to the parent directory (QPGMR, QTCM), and the owner of the parent directory (SANDERS), are granted the same private authority to the new directory.

Object ownership

This topic describes object ownership and its functions in the system.

Each object is assigned to 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 149 shows examples of how the system assigns ownership to new objects.

The owner of an object always has all the authorities for the object unless any or all authorities is removed specifically. As an object owner, you might choose to remove some specific authority as a precautionary measure provided you do not have *ALLOBJ special authority. For example, if a file exists that contains critical information, you might 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. The owner of a newly created integrated file system object has the same object authorities for that integrated file system object as the owner of the parent directory has to the parent directory. Check the Planning and setting up system security topic to see whether the rules for object authorities apply to all file systems or only to certain ones.

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 the group has members.

Note: Group ownership is a security risk as all members of the group obtain all authority and ownership rights to objects created by this user profile.

The following paragraphs apply to both library- and directory-based objects.

When changing an object's owner, you have the option to keep or revoke the former owner's authority.

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 on either performance or system operations, do not assign objects to only one owner profile for your entire IBM i environment. 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 98 for more information.

Group ownership of objects

This topic provides detailed information about the 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. Objects created into directories do not use the OWNER and GRPAUT values to determine ownership or group authority.

Note: Group ownership (OWNER=*GRPPRF) is a security risk as all members of the group obtain all authority and ownership rights to objects created by this user profile.

Note: Using the GRPAUT parameter with a value other than *NONE gives all other users who are members of the group profile that is specified on the GRPPRF parameter authority to objects created by this user. This may be a security risk.

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 149 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 includes *CREATE and the QAUDCTL system value includes *AUDLVL.

Related concepts

Group profiles

A *group profile* is a special type of user profile. Rather than giving authority to each user individually, you can use a group profile to define authority for a group of users.

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 might provide better performance than using 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 149 shows examples of how authority is assigned when new objects are created. For a directory-based object in some file systems, the object inherits the primary group of its parent directory. For example, if the parent directory has a primary group of FRED, then FRED will have problems trying to create anything in that parent directory. That is because the same profile cannot be both the owner and the primary group profile for the same object.

You can change the primary group for a library- or directory-based object using any of the following commands:

- Change Object Primary Group (**CHGOBJPGP**) command
- Change Primary Group (**CHGPGP**) command
- Option 9 on the Work with Objects by Primary Group (**WRKOBJPGP**) command

You can change the authority of the primary group using the Edit Object Authority (**EDTOBJAUT**) command or the grant and revoke authority commands. You can change the primary group's authority for a library- or directory-based object using the Change Authority (**CHGAUT**) command or the Work with Authority (**WRKAUT**) command.

Related concepts

Group profiles

A *group profile* is a special type of user profile. Rather than giving authority to each user individually, you can use a group profile to define authority for a group of users.

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.

The following situations 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 objects.

Assigning authority and ownership to new objects

You can assign authority and ownership to new objects on the system.

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 on page 150 through Figure 9 on page 153 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

You can assign adopted authority to a user program to allow the user to change a customer file.

Sometimes a user needs different authorities to an object or an application, depending on the situation. For example, a user might 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 to 1) give the user *USE authority to customer information to allow querying the files and 2) use adopted authority in the customer maintenance programs to allow the user to change the files.

When an object uses the owner's authority, this is called *adopted authority*. Objects of type *PGM, *SRVPGM, and *SQLPKG can adopt authority.

When you create a program, you specify a user profile (USRPRF) parameter on the CRTxxxPGM command. This parameter determines whether the program uses the authority of the owner of the program in addition to the authority of the user running the program.

Consult the Limit the use of adopted authority topic concerning security considerations and adopted authority when using SQL packages.

The following description 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 call stack. For example, assume that PGMA uses adopted authority:
  - If PGMA starts PGMB using the CALL command, these are the call stacks before and after the CALL command:

Table 120. Adopted authority and the CALL command

Call stack before CALL command:	**Call stack after CALL command:**
QCMD • • • PGMA	QCMD • • • PGMA PGMB

Because PGMA remains in the call 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 156 for more information about the USEADPAUT parameter.)

  - If PGMA starts PGMB using the Transfer Control (TFRCTL) command, the call stacks look like this:

Table 121. Adopted authority and the TFRCTL command

Call stack before TFRCTL command:	**Call stack after TFRCTL command:**
QCMD • • • PGMA	QCMD • • • PGMB

PGMB does not use the adopted authority of PGMA, because PGMA is no longer in the call 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 run 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.
- Adopted authority is not used by the integrated file systems, including the "root" (/), QOpenSys, QDLS, and user-defined file systems.
- 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 might 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 either on the command that creates the program (CRTxxxPGM) or on the Change Program (CHGPGM) or Change Service Program (CHGSRVPGM) 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 call 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 have been changed since the last time the command was run.

“Flowchart 8: How adopted authority is checked” on page 185 provides more information about adopted authority. The topic “Using adopted authority in menu design” on page 232 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.

Recommendations:

- Do not use an IBM supplied user profile as the owner of an application.
- Do not adopt authority of an IBM supplied user profile (don't use the IBM profile as the owner of the program that adopts).
- Set the LMTCPB(*YES) parameter on the user profile that is being used as the owner of the programs that adopt authority. This will prevent command line use if the user can break out of the application because of a programming error (security hole).

Adopted authority risks and recommendations

You should use adopted authorities with care to prevent possible security risks.

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 will 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 that these programs do not provide a means for the user to access objects outside the control of the program, such as command entry capability.
- Make sure that programs that adopt authority and call other programs perform library qualified calls. 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.

However, using adopted authority can also greatly increase the security of an application. For example, within an application you can set PUBLIC(*EXCLUDE) authority on all objects for the application. By using adopted authority while the application is running (using the authority of the application owning profile as the source of authority via program adopted authority) the user is authorized to the objects while the application is running. Once the application ends, the user no longer has authority to the application objects as the program adopted authority is no longer available. This technique prevents the user from accessing the data outside the application environment as the PUBLIC(*EXCLUDE) authority prevents access.

Programs that ignore adopted authority

You can specify the use adopted authority (USEADPAUT) parameter to control whether a program uses the adopted authority.

You might not want some programs to use the adopted authority of previous programs in the call stack. For example, if you use an initial menu program that adopts owner authority, you might 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 234 shows an example of how to use this parameter in menu design. See "Use Adopted Authority (QUSEADPAUT)" on page 36 for information about 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 might be useful when you do not know the USEADPAUT setting of the function that is called.

Related concepts

Ignoring adopted authority
The technique of using adopted authority in menu design 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.

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.

The primary use of an authority holder 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 descriptions apply 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.
- After 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

You should take security into consideration when using an authority holder.

An authority holder provides the capability of defining authority for a file before that file exists. Under certain circumstances, this can allow an unauthorized user to gain access to information. If a user knew that an application creates, moves, or renames a file, the user can create an authority holder for the new file. The user thus gains 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 topic describes commonly-used methods for setting up, maintaining, and displaying authority information about your system.

Appendix A, “Security commands,” on page 335 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

This section describes some characteristics of the displays that show object authorities.

Four displays show object authorities:

- Display Object Authority display
- Edit Object Authority display

- Display Authority display
- Work with Authority display

Figure 10 on page 159 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
*PUBLIC                 *EXCLUDE
PGMR1                   *ALL
DPTAR                   *CHANGE
DPTSM                   *USE
F3=Exit F11=Display detail object authorities F12=Cancel  F17=Top
```

Figure 10. 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
*PUBLIC                 *EXCLUDE    X
PGMR1                   *ALL        X    X     X      X     X
DPTAR                   *CHANGE     X
DPTSM                   *USE        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
*PUBLIC                 *EXCLUDE
PGMR1                   *ALL        X     X     X       X       X
DPTAR                   *CHANGE     X     X     X       X       X
DPTSM                   *USE        X                           X
```

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
*ADOPTED                USER DEF
*PUBLIC                 *EXCLUDE
PGMR1                   *ALL
*GROUP      DPTAR       *CHANGE
DPTSM                   *USE
```

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
*ADOPTED                USER DEF         X     X      X     X
*PUBLIC                 *EXCLUDE
PGMR1                   *ALL        X    X     X      X     X
*GROUP     DPTAR        *CHANGE     X
DPTSM                   *USE        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
*ADOPTED         USER DEF        X   X   X   X
*PUBLIC          *EXCLUDE
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
```

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)

Related information

System security tools

Working with libraries

You can specify the authority for libraries and new objects created in the 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 authorities:

- 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
*PUBLIC                  *AUTL
PGMR1                    *ALL
```

- 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  . . . . . :   TESTLIB

Type . . . . . :   PROD
ASP number . . . . . :   1
ASP device . . . . . :   *SYSBAS
Create authority . . . . . :   OBJLST
Create object auditing . . . . . :   *SYSVAL
Text description . . . . . :   Customer Rec
```

Creating objects

You can specify the authority of a new object.

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

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

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

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

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

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

Working with individual object authority

You can change the authority for an object.

To change the authority for an object, you must have one of the following authorities:

- *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, Work with Objects by Private Authority, Work with Objects by Primary Group, or Work with Objects display.

```
                           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
*PUBLIC                *AUTL
PGMR1                  *ALL
```

You can also use these commands to change object authority:

- 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

This topic provides information about specifying user-defined authorities.

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
*PUBLIC                  *AUTL
PGMR1                    USER DEF    X    X            X     X
```

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
*PUBLIC                  *AUTL
PGMR1                    USER DEF    X     X     X       X        X
```

Giving authority to new users

You can grant 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

You can also remove a user's authority for an object.

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.

Note: *EXCLUDE authority for a group profile can be overriden if the user has another group profile with private authority to the object.

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  Opr  Mgt  Exist  Alter  Ref
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

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

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

Working with authority for multiple objects

Learn how to make authority changes to more than one object at a time.

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 will 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

You can change the ownership of an object in several ways.

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

- 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 127 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

You can change the primary group or primary group's authority to an object.

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 authorities:

- *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 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 “Advantages of using an authorization list” on page 169 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 131 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.

Related tasks

Copying private authorities
You can copy the private authorities from one user profile to another using the Grant User Authority (GRTUSRAUT) command.

Working with authorization lists

This section introduces the steps for creating an authorization list.

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.

Advantages of using an authorization list

You can use authorization lists to protect objects on your system.

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, by default 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), ALWOBJDIF(*AUTL), or ALWOBJDIF(*COMPATIBLE) is specified on the restore command.
- 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 are secured by the list, there is still an advantage of using an authorization list over 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.

Creating an authorization list

Use the Create Authorization List (**CRTAUTL**) command to create 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       Name
Text 'description' . . . . . .   Files cleared at month-end

                           Additional Parameters

Authority  . . . . . . . . . .   *use           *CHANGE, *ALL, *USE, *EXCLUDE
```

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

Use the Edit Authorization List (EDTAUTL) display to give users authority to the authorization list you have created.

To work with the authority that users have for 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 143 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
*PUBLIC     *USE
PGMR1       *ALL        X
```

To give new users authority to the authorization list, press F6 (Add new users):

```
                               Add New Users

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

Type new users, press Enter.

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

Each user's authority to the list is actually stored as a private authority in that user's profile. You can also use commands to work with authorization list users, either interactively or in batch:

- Add Authorization List Entry (ADDAUTLE) to define authority for additional users.
- Change Authorization List Entry (CHGAUTLE) to change authority for users who are already authorized to the list.
- Remove Authorization List Entry (RMVAUTLE) to remove a user's authority to the list.
- Work with Authority (WRKAUT) to show the list of authorized users of an object.
- Change Authority (CHGAUT) to change a user's authority for the object.

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.

Use the Edit Object Authority display, the **GRTOBJAUT** command, the **WRKAUT** command, or the **CHGAUT** 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
*PUBLIC     *AUTL
PGMR1       *ALL
```

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 of the objects secured by 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
```

This is an information list only. You cannot add or remove objects from the list. You can also use the Display Authorization List Objects (**DSPAUTLOBJ**) command to view or print a list of all objects secured by the list.

Setting up an authorization list

The setup of an authorization list makes it easier to change who is authorized to the objects, and easier to secure any new objects with the same authorities as the existing objects.

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.

Follow these 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)
```

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.

Deleting an authorization list

You might also want to delete the authorization list that you have created.

You cannot delete an authorization list if it is used to secure any objects. Use the **DSPAUTLOBJ** command to list all of the objects secured by the list. Use either the Edit Object Authority display, Change Authority (**CHGAUT**), 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 153 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 might be accumulated to find sufficient authority for the object being accessed.

Authority checking flowcharts

This section introduces the flowcharts, 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 effect.

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 might 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 Figure 13 on page 177 (Flowchart 3: Check user authority).
- Step 6 in Figure 16 on page 183 (Flowchart 6: Group authority checking).
- Step 2 in Figure 19 on page 188 (Flowchart 8B: Checking adopted authority using private authorities).

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.

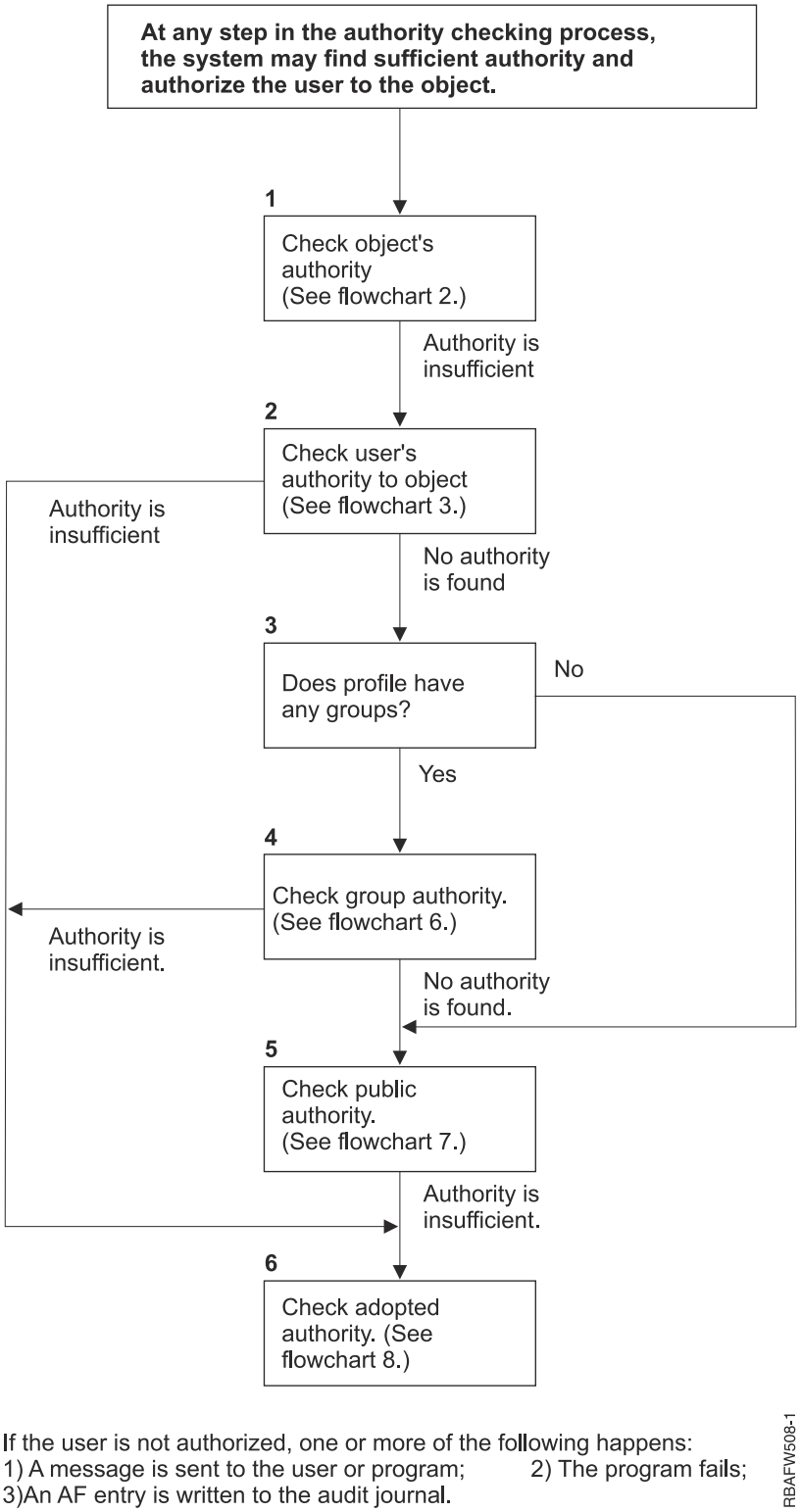

Figure 11. Flowchart 1: Main authority checking process

Description of Flowchart 1: Main authority checking process

Note: At any step in the authority checking process, the system might 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 proceeds to Step 5.
4. The system determines the group authority. (Refer to Flowchart 6). If the system determines that there is no group authority to the object, it proceeds to Step 5. If the system determines that the group authority to the object is not sufficient, 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.)

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.

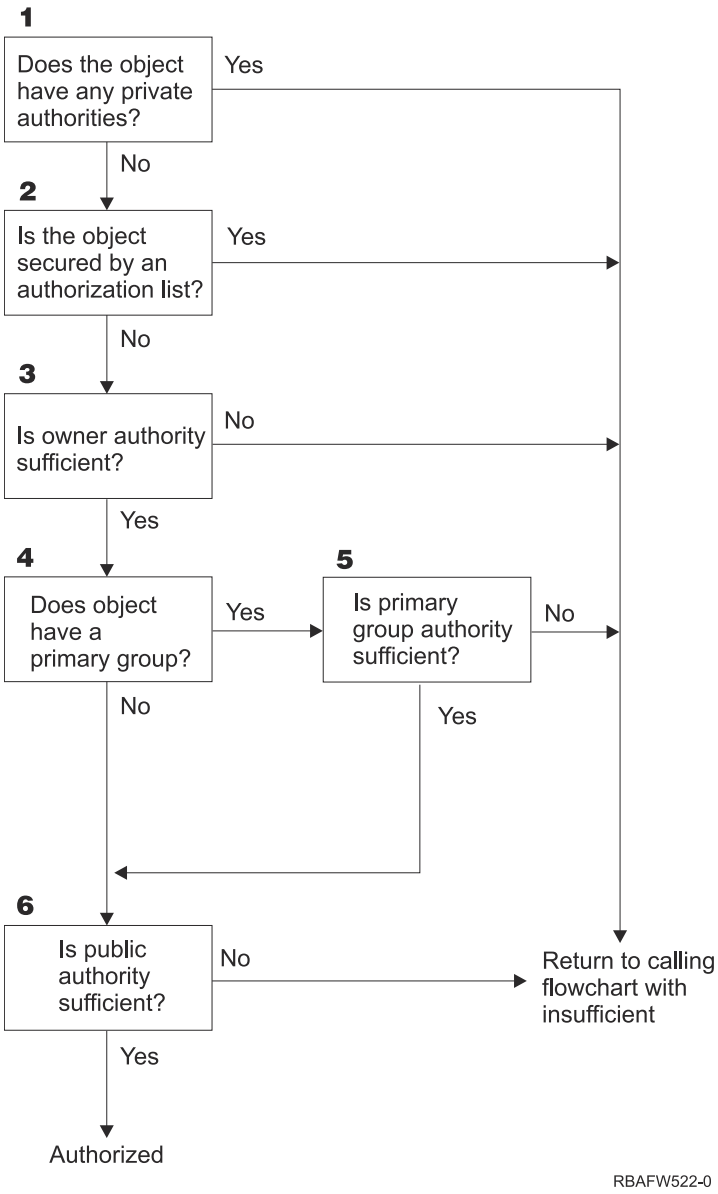

Figure 12. Flowchart 2: Fast path for object authority

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

Flowchart 3: How user authority to an object is checked

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

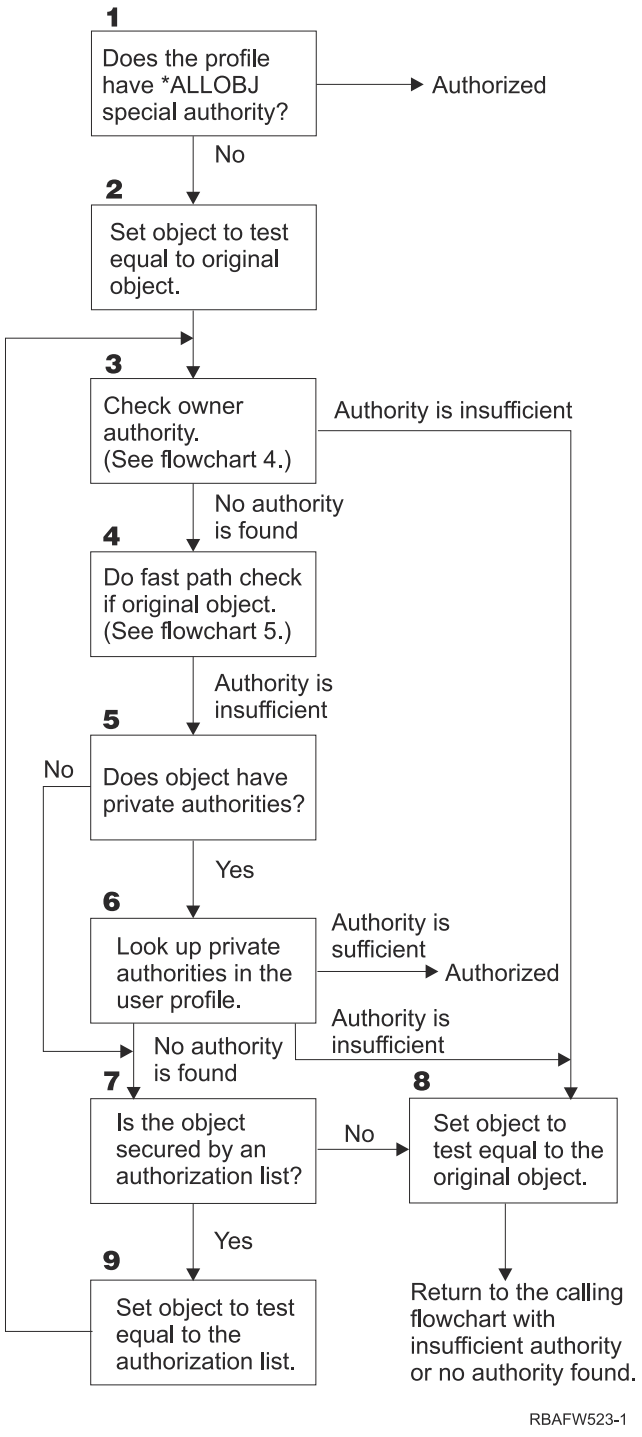

Figure 13. Flowchart 3: Check user authority

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 checks the owner authority. If the authority is insufficient, 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 checks 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

Flowchart 4 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.

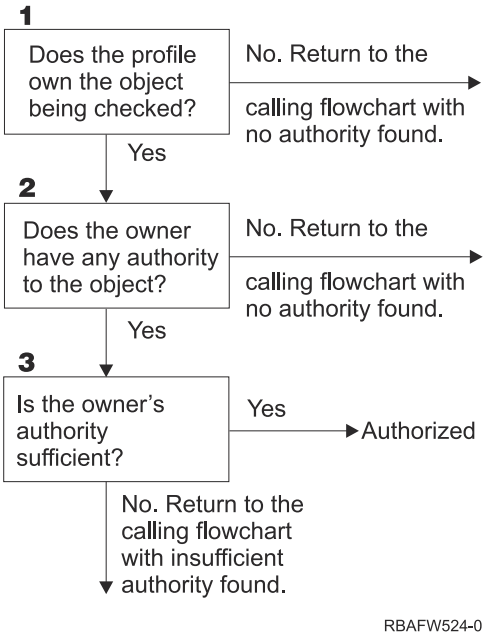

Figure 14. Flowchart 4: Owner authority checking

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 it 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 the owner has authority to the object, 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 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

Flowchart 5 shows the fast path for testing user authority without searching private authorities.

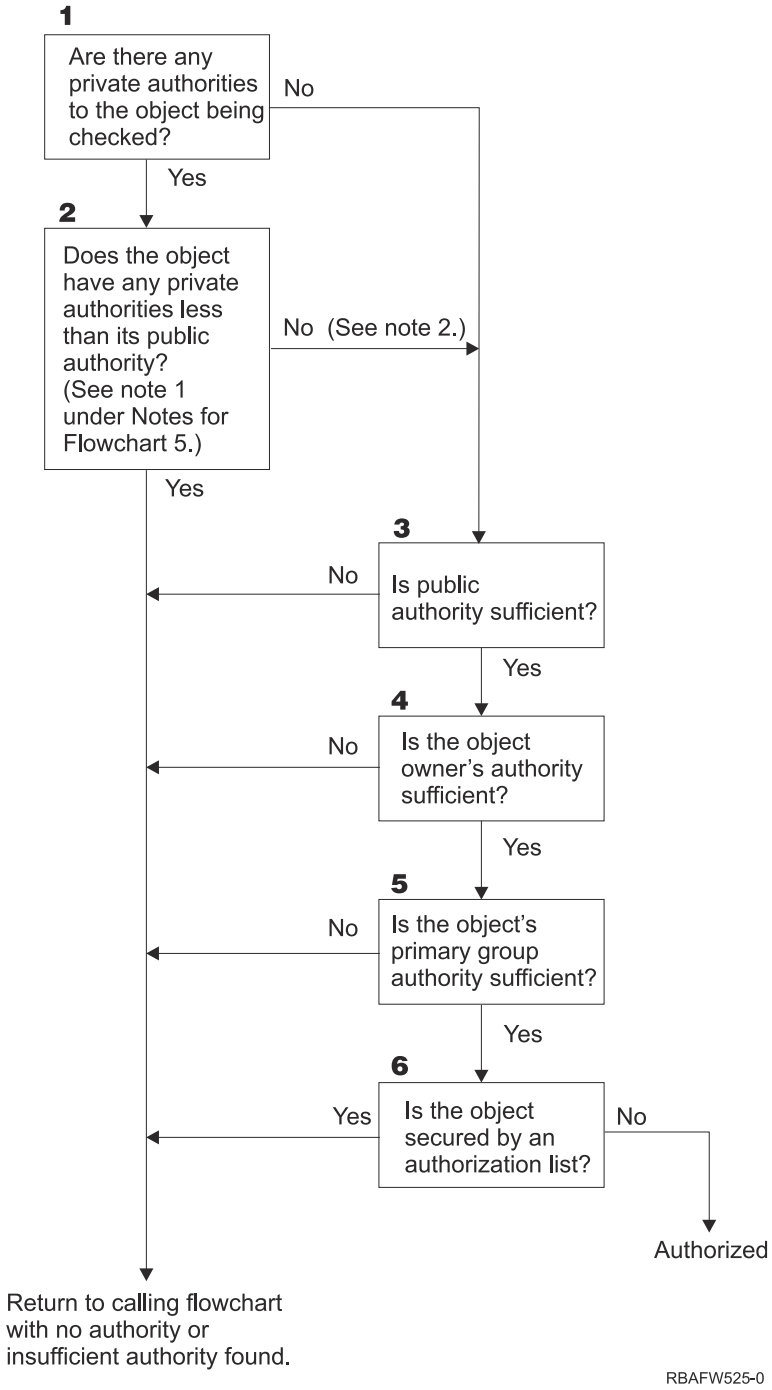

Figure 15. Flowchart 5: Fast path for user authority

Flowchart 5 notes:

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 122 on page 181, 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 122. 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 any private authorities or 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 might be a member of up to 16 groups. A group might have private authority to an object, or it might be the primary group for an object.

Authority from one or more of the user's groups might 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 123 on page 182 shows the authorities for the CRLIM file:

Table 123. 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 Figure 16 on page 183 shows the steps in checking group authority.

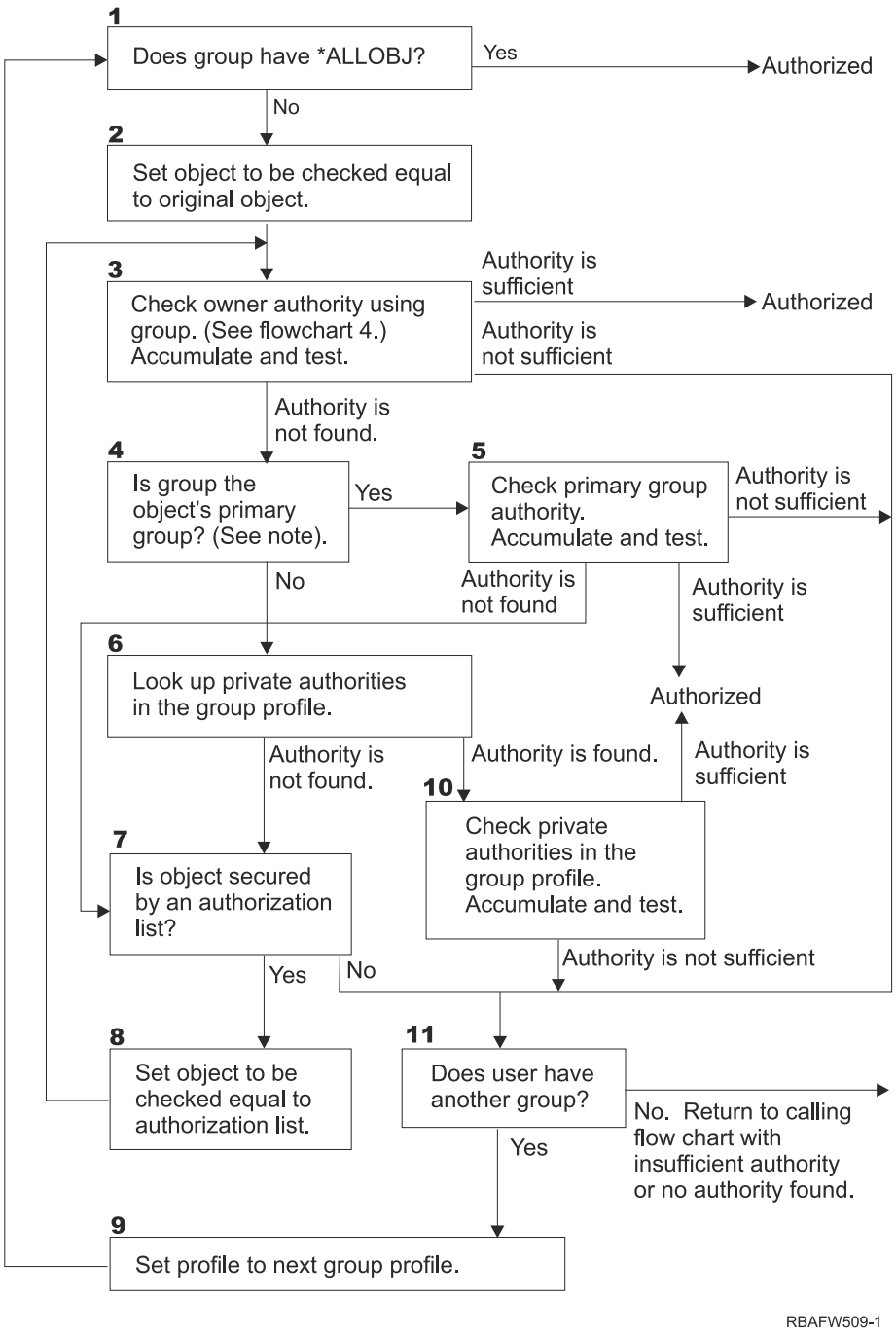

Figure 16. Flowchart 6: Group authority checking

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. The group does not have *ALLOBJ authority so 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 11. If the authority is not found, then the authority check proceeds to Step 4.

4. The owner authority is not found so 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. The group is the object's primary group so 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 not found, then the authority check goes to Step 7. If the primary group authority is insufficient, then the authority check goes to Step 11
6. The group is not the object's primary group so 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. No authority is found for the private authorities for the group profile so 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. The object is secured by an authorization list so the system set the object to be checked equal to the authorization list and authority check returns to Step 3.
9. The user belongs to another group profile so the system sets the profile to the next group profile and returns to Step 1 to start the authority checking process over again.
10. Authority is found for private authorities within the group profile so 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 11.
11. Authority is not found or is insufficient so 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:

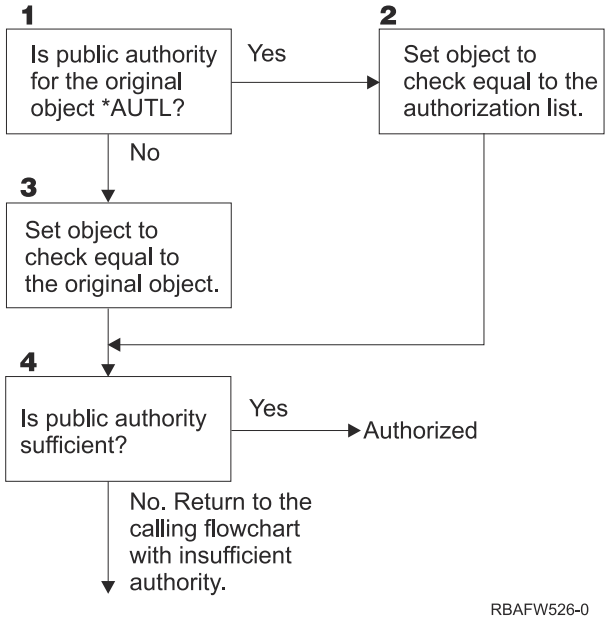

Figure 17. Flowchart 7: Check public authority

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 if 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 might use adopted authority from the original program the user called or from earlier programs in the call 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 18 on page 186 and Figure 19 on page 188 show the process for checking adopted authority.

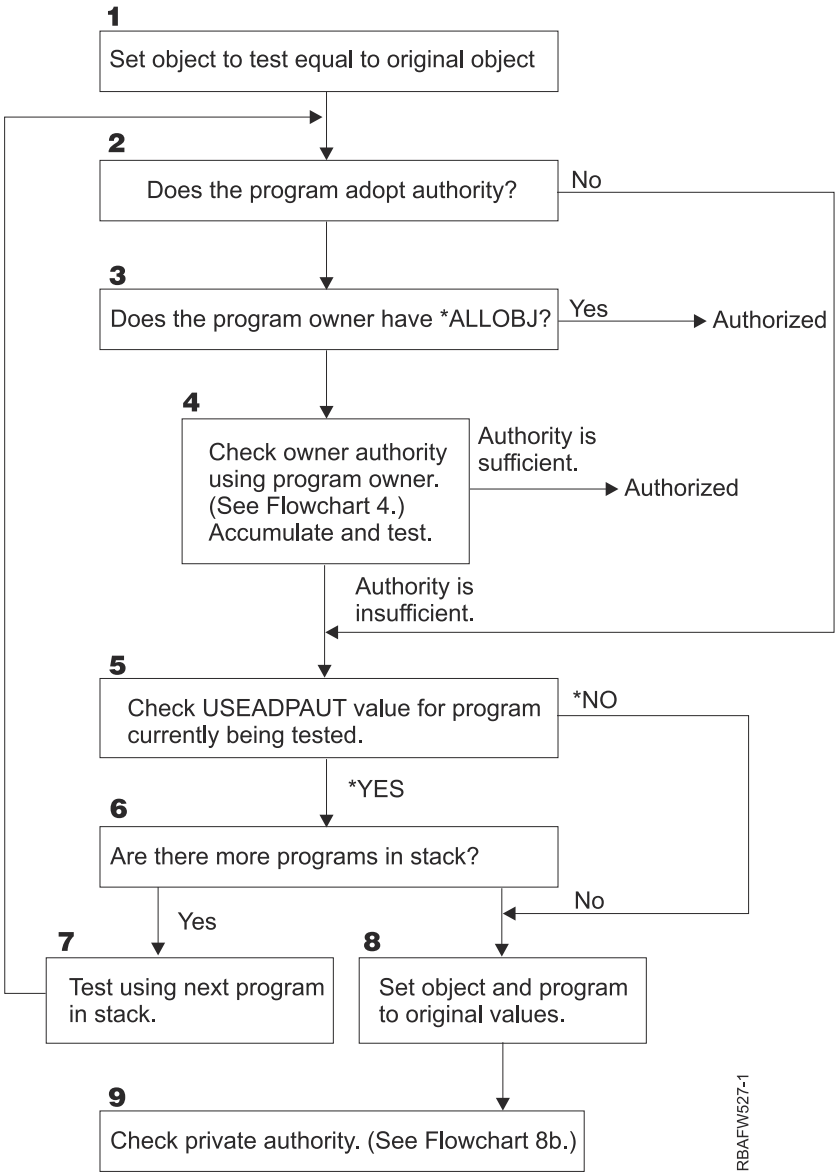

*Figure 18. Flowchart 8A: Checking adopted authority user *ALLOBJ and owner*

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. Test using the next program in the stack and start back at Step 2.
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 checks private authority. This is described in Flowchart 8B: Checking adopted authority using private authorities.

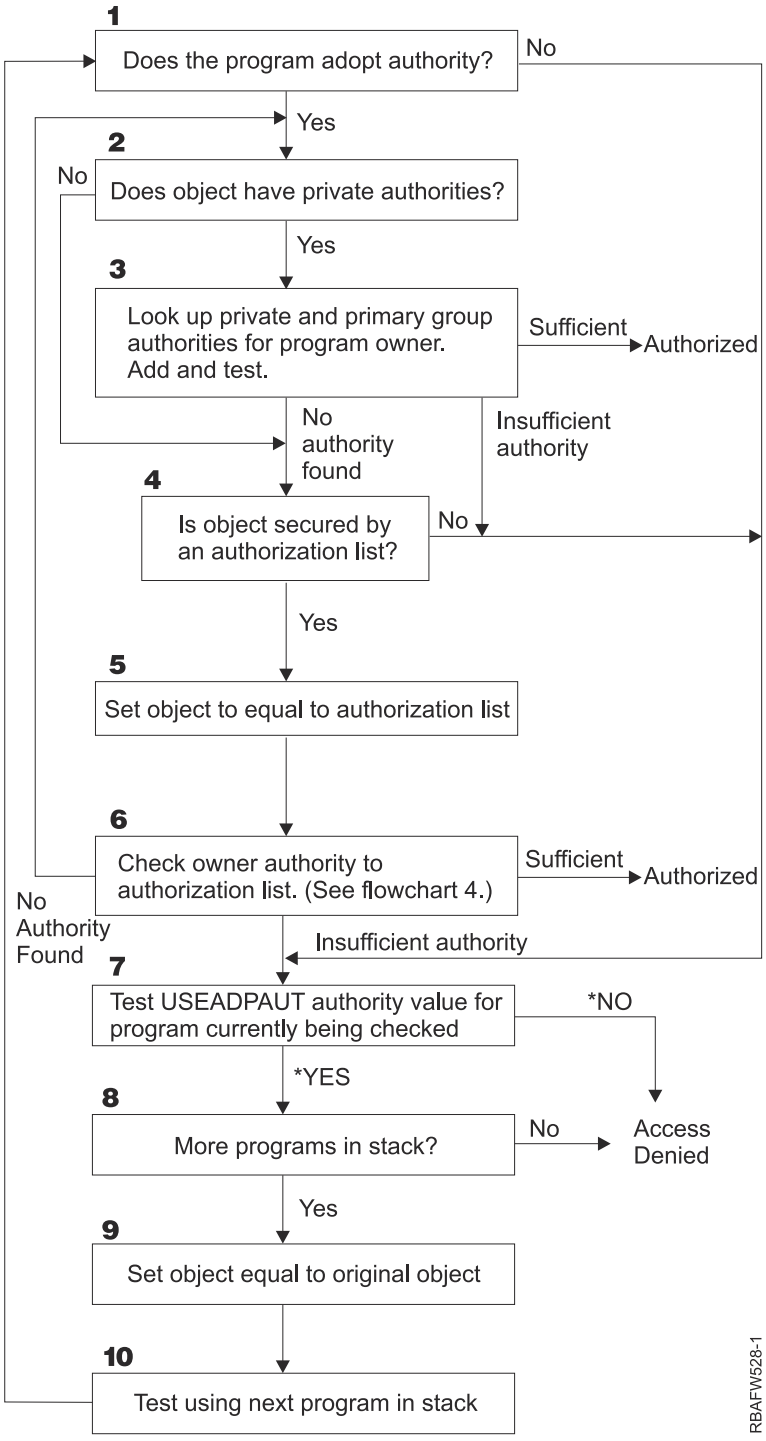

Figure 19. Flowchart 8B: Checking adopted authority using private authorities

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. Test using next program in stack and start back at Step 1.

Related concepts

Ignoring adopted authority

The technique of using adopted authority in menu design 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.

Authority checking examples

This section includes 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 might occur.

Figure 20 on page 189 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 20. Authority for the PRICES file

Case 1: Using private group authority

This case demonstrates how to use 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.

i) Flowchart 4, step 1. Return to Flowchart 3 with no authority found. ROSSM does not own the PRICES file.

c) Flowchart 3, step 4.

i) 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.

i) 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 typically causes two searches of private authorities (for the user and the group), when public authority is not adequate. One search of the private authority can be avoided by making DPTSM the primary group for the PRICES file.

Case 2: Using primary group authority

This case demonstrates how to use 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 21 on page 190 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 21. 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.

   i) Flowchart 4, step 1. ANDERSJ does not own the CREDIT file. Return to Flowchart 3 with no authority found.

c. Flowchart 3, step 4.

   i) Flowchart 5, step 1. The CREDIT file has no private authorities.

   ii) 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.

      i) 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.

Related concepts

Considerations for 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.

Case 3: Using public authority

This case describes the steps of 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:

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

This case describes how to use 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.
      i) Flowchart 4, step 1. JONESP does not own the PRICES file. Return to Flowchart 3 with no authority found.
   c. Flowchart 3, step 4.
      i) Flowchart 5, steps 1, 2, and 3. Public authority is sufficient.
      ii) Flowchart 5, step 4. Owner authority is sufficient. (OWNCP has *ALL.)
      iii) Flowchart 5, step 5. The PRICES file does not have a primary group.
      iv) 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, which are less than public authority, for an object. 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

This case demonstrates the performance advantage in 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.
      i) Flowchart 4, step 1. SMITHG does not own the PRICES file. Return to Flowchart 3 with no authority found.
   c. Flowchart 3, step 4.
      i) 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.

      i) 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 effect 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 189.

- 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 is the same. However, having a group profile own application objects might 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 can 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 is the same as Case 5.

Case 6: User and group authority

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

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

   c. Flowchart 3, step 4.

      i) 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.

   i) Flowchart 8B, step 1. Program CPPGM01 does not adopt authority.

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

Analysis:

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 might 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

This case demonstrates the performance advantage of using 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 22. 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:

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

This case shows the advantage of using 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.

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

c. Flowchart 3, step 4.

   i) 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.

      i) 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

This case demonstrates the advantage of using authorization lists.

The ARWKR01 file in library CUSTLIB is secured by the ARLST1 authorization list. Figure 23 on page 195 and Figure 24 on page 195 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 23. 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 24. 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.
      i) Flowchart 4, step 1. AMESJ does not own the ARWRK01 file. Return to Flowchart 2 with no authority found.
   c. Flowchart 3, step 4.
      i) 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.
      i) 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 will add additional steps to this example, but it will 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 197.

Case 10: Using multiple groups

This is an example of 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 25 on page 196 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 25. 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.
      i) Flowchart 4, step 1. WOODBC does not own the CRLIM file. Return to Flowchart 3 with no authority found.
   c. Flowchart 3, step 4.
      i) 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.
      i) 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

This case shows a poor authority design.

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 26 on page 197 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 26. Authority for CRLIMWRK file

The CRLIMWRK file is secured by the CRLST1 authorization list. Figure 27 on page 198 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 27. 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.
        i) Flowchart 4, step 1. WAGNERB does not own the CRLIMWRK file. Return to Flowchart 3 with no authority found.
    c. Flowchart 3, step 4.
        i) 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.
        i) 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.
        i) 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.

i) 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.

i) 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.

i) 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.

i) 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 might 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 changes the performance characteristics of this example. Assume that DPTAR is WAGNERB's first group profile (GRPPRF). The system searches 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

The system creates authority caches for users to provide flexibility and performance enchancement.

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 effect on system performance. This is especially true if users access the same objects repeatedly.

Chapter 6. Work management security

This section discusses security issues associated with work management on the system.

The following issues are described in this section.

Related information
Work management

Job initiation

The system checks the authority to some objects when a job is started.

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 to some of these objects is checked before the job is allowed to start, while authority to other objects is checked after the job starts. Inadequate authority might cause errors or may cause the job to end.

Objects that are part of the job structure for a job can be specified in the job description, the user profile, and on the Submit Job (SBMJOB) command for a batch job.

Starting an interactive job

This topic 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 202 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

This topic includes 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 or 50 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

You can change the parameters for a batch job when it is running under adopted authority.

When a new job is started, a new call stack is created for the job. Adopted authority cannot take effect until the first program is added to the call stack. Adopted authority cannot be used to gain access to any objects, such as an output queue or a job description, which 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 Job commands for the authority that is required to change parameters for a job.

Workstations

The system performs authority checking for a workstation when you sign on.

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 28 on page 203 shows the logic for determining whether a user is allowed to sign on at a device:

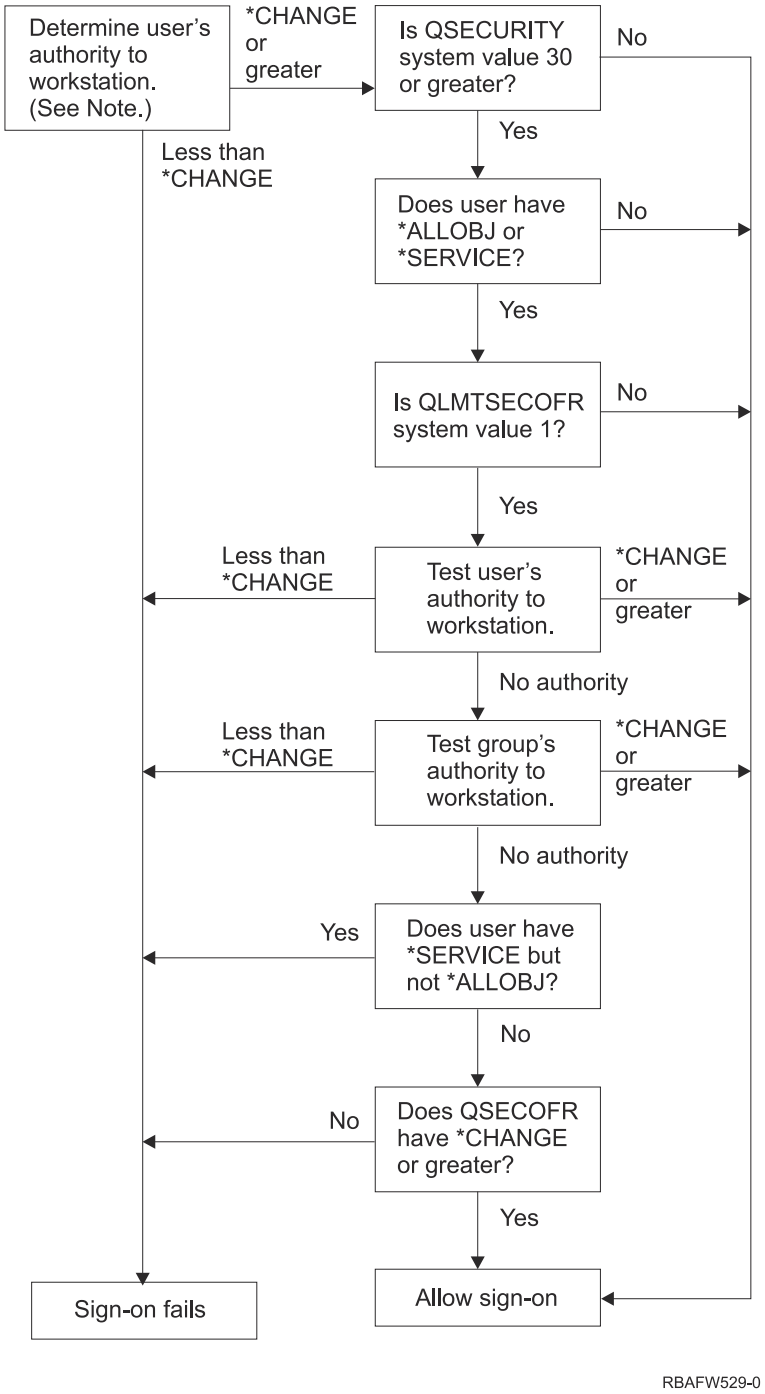

Figure 28. Authority checking for workstations

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

- *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 call 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, the sign-on fails. If the authority is *CHANGE or greater, the system checks if 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 authority. 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's group has no authority to the workstation, the system checks whether 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 has *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

You can specify the ownership of device descriptions to control the authority to the devices.

The default public authority on the CRTDEVxxx commands is *CHANGE. Devices are created in the 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.

When changing the signon screen display file, the system administrator must make sure not to change the field names or buffer lengths of the display file when adding text to the display file. Changing the field names or buffer lengths can cause signon to fail.

Changing the signon screen display

You can change the source code for the signon display file to change the 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

You need to copy the appropriate source file to create your own signon screen display.

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 IBM i 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

This topic includes the steps for changing the signon display file.

To change the format of the Signon display, perform the following steps:

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 to the application.
   - 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 is 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, perform the following steps:
   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

The subsystem descriptions perform several functions on the system.

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.

Related concepts

Signing on without a user ID and password
Your security level determines how the system controls signing on without a user ID and password.

Controlling how jobs enter the system

You can use the subsystem descriptions to control 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 signon is attempted and the auditing function is active. At security level 40 and higher, the system does not permit default signon, even if a combination of workstation entry and job description exists that allows 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 215 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 206 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 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

You can specify the authorities to control access to the system operator message queue

The IBM i 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 might 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 Message commands.) 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 124 on page 208 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 124. 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.

Related concepts

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.

Planning libraries
A library is like a directory used to locate the objects in the library. Many factors affect how you choose to group your application information into libraries and manage libraries.

Security risks of library lists

This topic gives specific examples of the possible security exposures of library lists and how to avoid them.

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 might be able to perform functions that break your security requirements.

"Library security and library lists" on page 140 provides some general information about the issues associated with library lists.

This section provides two examples of how changes to a library list might break security requirements.

Change in function

This example shows the possible risk of a change in function when calling a program in the library.

Figure 29 on page 209 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.

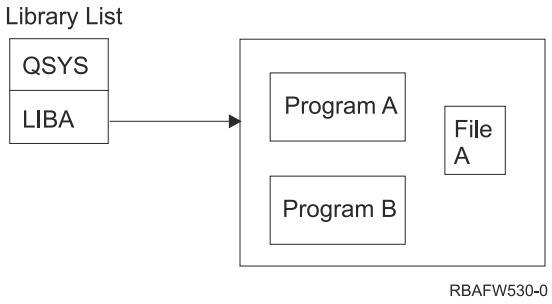

Figure 29. Library list–expected environment

A programmer or another knowledgeable user might 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:

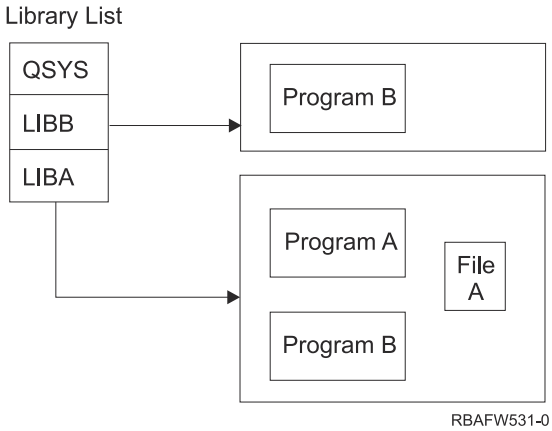

Figure 30. Library list–actual environment

Unauthorized access to information

The example demonstrates the potential risk of unauthorized access to information in the library.

Assume Program A in Figure 29 on page 209 adopts the authority of USER1, who has *ALL authority to File A. Assume that Program B is called by Program A (adopted authority remains in effect). A knowledgeable user can create a substitute Program B that just calls the command processor. The user will have a command line and complete access to File A.

Recommendations for system portion of library list

This topic provides the recommendations for the system portion of the 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 229.

Recommendations for product library

In this topic you will find the recommendations for protecting the product library.

The product library portion of the library list is searched before the user portion. A knowledgeable user can 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

This topic provides the recommendations to ensure the security of your system when using 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 228 to set the current library during application processing.

Recommendations for the user portion of the library list

In this topic you will find the recommendations for controlling the user portion of the library list.

The user portion of the library list often 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.

Here are some suggested alternatives for controlling the user portion of the library list to make sure that 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 230 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 required 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 required for the application. When the application ends, return the library list to its original setting. See "Controlling the user library list" on page 228 for an example of this technique.

Printing

You can control the security of the output queues on your system.

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 Advanced Function Presentation topic provides examples of how to direct output to a particular output queue.

Securing spooled files

You can specify several parameters to control the security of a spooled file.

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 to 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

You can specify the Display Data (DSPDTA) parameter to protect the contents of a spooled file.

The DSPDTA parameter 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 authorities: • *JOBCTL special authority if the OPRCTL parameter is *YES. • *READ, *ADD, and *DLT 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

You can use the Authority to Check (AUTCHK) parameter to control a user's authority to change or delete a spooled file on your system.

The AUTCHK parameter determines whether *READ, *ADD, and *DLT 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 Operator Control (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

This topic includes the reference information about the output queue parameters and authorities required for performing printing management functions.

Table 125 on page 213 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 561.

The authority and output queue parameters for all commands associated with spooled files are listed on "Spooled file commands" on page 543. Output queue commands are listed on "Output queue commands" on page 511.

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. Make careful consideration when giving *SPLCTL special authority to any user.

Table 125. Authority required to perform printing functions

Printing function	**Output queue parameters**			**Output queue authority**	**Special authority**
	DSPDTA	**AUTCHK**	**OPRCTL**		
Add spooled files to queue [1]				*READ	None
			*YES		*JOBCTL
View list of spooled files (WRKOUTQ command [2])				*READ	None
			*YES		*JOBCTL
Display, copy, or send spooled files (DSPSPLF, CPYSPLF, SNDNETSPLF, SNDTCPSP[2])	*YES			*READ	None
	*NO	*DTAAUT		*READ, *ADD, *DLT	None
	*NO	*OWNER		Owner [3]	None
	*YES		*YES		*JOBCTL
	*NO		*YES		*JOBCTL
	*OWNER				

Table 125. Authority required to perform printing functions (continued)

Printing function	Output queue parameters			Output queue authority	Special authority
	DSPDTA	**AUTCHK**	**OPRCTL**		
Change, delete, hold, and release spooled file (CHGSPLFA, DLTSPLF, HLDSPLF, RLSSPLF [2])		*DTAAUT		*READ, *ADD, *DLT	None
		*OWNER		Owner [3]	None
			*YES		*JOBCTL
Change, clear, hold, and release output queue (CHGOUTQ, CLROUTQ, HLDOUTQ, RLSOUTQ [2])		*DTAAUT		*READ, *ADD, *DLT	None
		*OWNER		Owner [3]	None
			*YES		*JOBCTL
Start a writer for the queue (STRPRTWTR, STRRMTWTR [2])		*DTAAUT		*CHANGE	None
			*YES		*JOBCTL

1 This is the authority required to direct your output to an output queue.

2 Use 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

These examples demonstrate how to set 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.

Client Request Access (PCSACC) network attribute

The PCSACC network attribute determines how the IBM i 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 IBM i platform, but it doesn't control 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 IBM i Access clients.

Possible values for PCSACC:	
*REJECT	IBM i Access rejects every request from the personal computer to access objects on the IBM i platform. An error message is sent to the PC application.
*OBJAUT	The IBM i 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 IBM i 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 IBM i 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

Use the instructions in this topic to protect the files on your system.

Normal security measures on your system might not be sufficient protections if the IBM i 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 IBM i 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 a IBM i 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 IBM i 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 IBM i Access licensed program. Using an exit program to verify all requests is the only adequate protection measure.

The IBM i 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

Related information

Programming: IBM i Access

DDM Request Access (DDMACC) network attribute

The DDM Request Access (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 functions involving DDM files and DRDA connection requests. 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.

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 might 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 restrict the save and restore operations to protect your system.

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 and optical 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" on page 217 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

This topic shows an example of restricting the save and restore commands.

You can follow these steps 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 example:

```
CRTAUTL AUTL(SRLIST) TEXT('Save and Restore List')
AUT(*EXCLUDE)
```

2. To use the authorization list to secure the save commands, type the following example:

```
GRTOBJAUT OBJ(SAV*) OBJTYPE(*CMD) AUTL(SRLIST)
```

3. To ensure *PUBLIC authority comes from the authorization list, type the following example:

```
GRTOBJAUT OBJ(SAV*) OBJTYPE(*CMD) USER(*PUBLIC)
   AUT(*AUTL)
```

4. To use the authorization list to secure the restore commands, type the following example:

```
GRTOBJAUT OBJ(RST*) OBJTYPE(*CMD) AUTL(SRLIST)
```

5. To ensure *PUBLIC authority comes from the authorization list, type the following example:

```
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 might 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 them with an 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)
GRTOBJAUT OBJ(QSRRSTO) OBJTYPE(*PGM) AUTL(SRLIST)
GRTOBJAUT OBJ(QSRRSTO) OBJTYPE(*PGM) 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 action:

```
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 might want to run certain reports or program compiles in batch. A job running in batch often 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 action:

- 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

This section contains guidelines to help application developers and systems managers include security as part of the overall design. It also contains examples of techniques that you can use to accomplish security objectives on your system.

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.

Some of the examples in this section 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 Plan and set up system security 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 might find it useful to review the forms and examples in the Plan and set up system security topic 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 Plan and set up system security topic in the information center also uses a set of example applications for a fictional company called the JKL Toy Company. This section discusses design considerations for the same set of example applications. Figure 31 on page 221 shows the relationships between user groups, applications, and libraries for the JKL Toy Company:

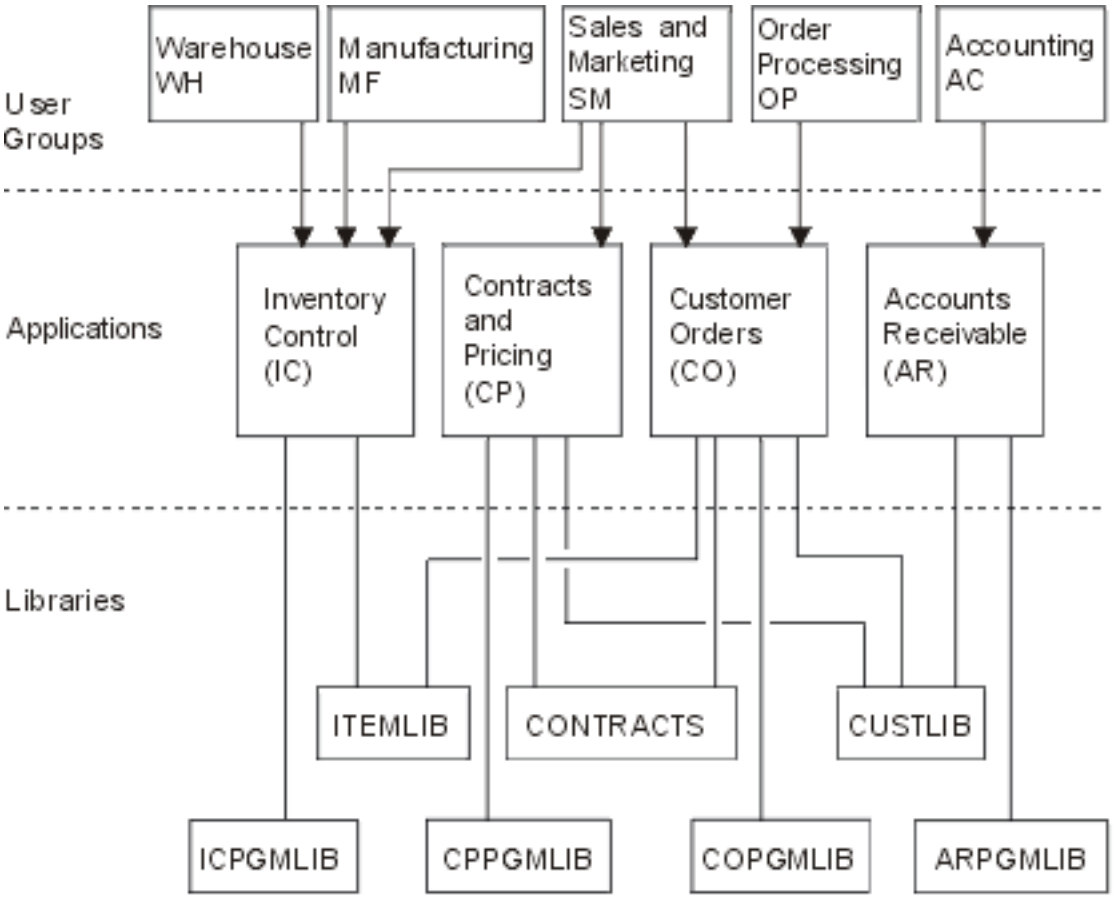

Figure 31. Example applications

Description of graphic

This graphic shows 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. These user groups have different accesses to different applications, which are stated in the following list.

- The Warehouse, Manufacturing and Sales and Marketing user groups can all 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 can also access the Customer Order application.
- The Accounting user group only has access to the Accounts Receivable application.

Related information

Scenarios for HTTP Server

Overall recommendations for security design

Keeping your security design as simple as possible makes it easier to manage and audit security. It also improves application performance and backup performance.

Here 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: If you use a product such as IBM i Access or if you have communication lines attached to your system, do not rely only on limiting capabilities in the user profile and menu access control. You must use resource security to secure any objects that you do not want to be 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 and directories. 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.

Related concepts

Resource security

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

Planning password level changes

Changing password levels should be planned carefully. Operations with other systems might fail or users might not be able to sign on to the system if you haven't planned for the password level change adequately.

Before changing the QPWDLVL system value, make sure that 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, even 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, might 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 that a user enters on a sign-on screen, must be upgraded to work with the 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 algorithm for QPWDLVL 2 or 3, even if the specific characters typed in are correct, will not be accepted. This also applies to any IBM i to IBM i 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 (such as IBM Toolbox for Java) 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 you chang the QPWDLVL system value.

Considerations for changing QPWDLVL from 0 to 1

Password level 1 allows a system, which doesn't need to communicate with the IBM i Support for Windows Network Neighborhood (IBM i NetServer), to eliminate the IBM i NetServer LAN manager passwords. IBM i NetServer LAN manager passwords only affect Windows 95/98/ME clients. The LAN manager passwords have been disabled by Windows since Vista so removing them will not affect current versions of Windows. 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 IBM i NetServer LAN manager password.

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

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.

Before changing QPWDLVL to 2, the system administrator should use the **PRTUSRPRF TYPE(*PWDLVL)** command to locate all of the user profiles that do not have a password that is usable at password level 2. Depending on the profiles located, the administrator can 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 meets the following description, 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.

- The 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.
- The user signs on through a product that sends clear text passwords.

Subsequent signons will be validated against the password level 2 passwords.

Any client that uses password substitution will not work correctly at QPWDLVL 2 if the client hasn't been updated to use the new password (passphrase) substitution scheme. The administrator should check whether a client which hasn't been updated to the new password substitution scheme is required.

The clients that use password substitution include:

- TELNET
- IBM i Access Client Solutions
- IBM i Host Servers
- QFileSrv.400
- IBM i NetServer Print support
- DDM
- DRDA
- SNA LU6.2

It is highly recommended that the security data be saved before changing to QPWDLVL 2. This can help make the transition back to QPWDLVL 0 or 1 easier if that becomes necessary.

Avoid changing password system values, such as QPWDMINLEN, QPWDMAXLEN, and QPWDRULES, until after you have tested QPWDLVL 2. This makes 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 allows QPWDLVL to be changed to 2. Therefore, if you use a password validation program, you might want to write a new one that can be registered for the QIBM_QSY_VLD_PASSWRD exit point, format VLDP0100, by using the ADDEXITPGM command.

IBM i NetServer LAN manager passwords are still supported at QPWDLVL 2, so any function/service that requires an IBM i NetServer LAN manager password should still function correctly.

After you are comfortable with running the system at QPWDLVL 2, you can change the password system values to use longer passwords. However, you need to be aware that longer passwords have these effects:

- If passwords greater than 10 characters are specified, the password level 0 and 1 password is cleared. This user profile will not be able to sign on 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 IBM i NetServer LAN manager password for the user profile is cleared. The LAN manager password is used to communicate with IBM i Support for Windows Network Neighborhood (IBM i NetServer) product and only affects Windows 95/98/ME clients. The LAN manager passwords have been disabled by Windows since Vista so removing them will not affect current versions of Windows.
- 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 IBM i NetServer LAN manager password values (if generated).

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

Considerations for changing QPWDLVL from 2 to 3

After running the system at QPWDLVL 2 for some period of time, you can consider moving to QPWDLVL 3 to maximize the password security protection.

At QPWDLVL 3, all IBM i NetServer LAN manager passwords are cleared so a system should not be moved to QPWDLVL 3 until there is no need to use IBM i NetServer LAN manager passwords. LAN manager passwords are used to communicate with IBM i Support for Windows Network Neighborhood (IBM i NetServer) product and only affects Windows 95/98/ME clients. The LAN manager passwords have been disabled by Windows since Vista so removing them will not affect current versions of Windows.

At QPWDLVL 3, all password level 0 and 1 passwords are cleared. The administrator can use the **DSPAUTUSR** or **PRTUSRPRF** command to locate user profiles which don't have password level 2 or 3 passwords associated with them.

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

Changing QPWDLVL 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 might be cases where a lower QPWDLVL value must be reinstated.

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

Considerations for changing from QPWDLVL 3 to 2

This change is relatively easy. After the QPWDLVL is set to 2, the administrator needs to determine if any user profile is required to contain IBM i NetServer LAN manager 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 might need to be changed back to values compatible with IBM i NetServer LAN manager passwords 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 (such as 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

Before changing QPWDLVL to 1, you should use the DSPAUTUSR or PRTUSRPRF TYPE(*PWDINFO) command to locate any user profiles that do not have a password level 0 or 1 password. If the user profile requires a password after the QPWDLVL is changed, make sure 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 causes 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 underline.
  - The password does not begin with a numeric or underline character.

  For example, changing the password to a value of RainyDay can result in the system generating a password level 0 and 1 password of RAINYDAY. But changing the password value to Rainy Days In April can 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 cannot 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 IBM i NetServer LAN manager 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 those for changing from QPWDLVL 2 to 1 except that all IBM i NetServer LAN manager passwords are retained when the change takes effect.

Considerations for changing from QPWDLVL 1 to 0

After changing QPWDLVL to 0, you should use the DSPAUTUSR or PRTUSRPRF command to locate any user profiles that do not have an IBM i NetServer LAN manager password. If the user profile requires an IBM i NetServer LAN manager password, it can be created by changing the user's password or signing on through a mechanism that presents the password in clear text.

You can then change QPWDLVL to 0.

Planning libraries

A library is like a directory used to locate the objects in the library. Many factors affect how you choose to group your application information into libraries and manage libraries.

Library security is effective only if the rules below 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.

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.

*USE authority to a library allows you 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.

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.

Here are two examples of using library security for the JKL Toy Company applications. (See Figure 31 on page 221 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 (*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 this information. 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 an SQL 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.

Related concepts

Library security
You can use library security to protect information.

Related reference

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.

Related information

Scenarios for HTTP Server

Planning applications to prevent large profiles

To reduce impacts on the performance and security of your system, you need to plan your applications carefully to avoid large profiles.

Because of the potential impacts on performance and security, perform the following actions to prevent 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 owning 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 affect 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 create an authorized object entry for each user with authority to the authorization list.

Library lists

The library list for a job represents a security exposure, while it provides flexibility. 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.

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 might prevent you from using qualified names. If your applications rely on library lists, the following techniques can reduce the security exposure.

Note: By using the code examples, you agree to the terms of the Chapter 11, “Code license and disclaimer information,” on page 333.

Controlling the user library list

As a security precaution, you might want to make sure that 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 list that you want, and restore it at the end of the application.

Here is a sample program to do this:

Note: By using the code examples, you agree to the terms of the Chapter 11, “Code license and disclaimer information,” on page 333.

```
           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 32. 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. This is 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

You might also need to change the system portion of the library list to protect your system.

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 32 on page 228, 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 value that you want.
- 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 actions:
  - 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 that 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 33 on page 230 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 33. 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 that 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 232 gives an example of how to do this.
- Consider controlling access to the System Request Menu or some of the options on this menu.
- 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.

For example, 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 34. Sample inquiry menu

Note: By using the code examples, you agree to the terms of the Chapter 11, “Code license and disclaimer information,” on page 333.

Related concepts

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. This might represent a security exposure because the public authority to the System Request Menu is *USE when a system is shipped.

Related reference

Limit capabilities
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 a tool for preventing users from experimenting on the system.

Related information

Scenarios for HTTP Server

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.

Examples of the type of information that a security administrator needs are:

- 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 35 on page 231 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 35. Format for menu security requirements

Using adopted authority in menu design

The availability of decision-support tools, such as Query/400, poses challenges for security design. 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.

For example, you might 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.

Note: “Objects that adopt the owner's authority” on page 153 describes how adopted authority works. “Flowchart 8: How adopted authority is checked” on page 185 describes how the system checks for adopted authority.

Figure 36 on page 232 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 36. 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. Here is an example of the Inventory Control program (ICSTART).

Note: By using the code examples, you agree to the terms of the Chapter 11, “Code license and disclaimer information,” on page 333.

```
PGM
ADDLIBLE ITEMLIB
ADDLIBLE ICPGMLIB
GO ICMENU
RMVLIBLE ITEMLIB
RMVLIBLE ICPGMLIB
ENDPGM
```

Figure 37. 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 38 on page 232 shows the QRYSTART program:

```
PGM
ADDLIBLE ITEMLIB
ADDLIBLE CUSTLIB
STRQRY
RMVLIBLE ITEMLIB
RMVLIBLE CUSTLIB
ENDPGM
```

Figure 38. Sample program for query with adopted authority

The menu system uses three types of user profiles, shown in Table 126 on page 233. Table 127 on page 233 describes the objects used by the menu system.

Table 126. 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	Not applicable	As needed by application	Not applicable
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	Not applicable	None	Not applicable

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 127. Objects used by menu system

Object name	Owner	Public authority	Private authorities	Additional information
MENU1 in QGPL library	See Note	*EXCLUDE	*USE authority for any users who are allowed to use the menu	In QGPL library because users do not have authority to application libraries
ICSTART program in QGPL	OWNIC	*EXCLUDE	*USE authority for users authorized to Inventory Control application	Created with USRPRF(*OWNER) to adopt OWNIC authority
QRYSTART program in QGPL	QRYUSR	*EXCLUDE	*USE authority for users authorized to create or run queries	Created with USRPRF(*OWNER) to adopt QRYUSR authority
ITEMLIB	OWNIC	*EXCLUDE	QRYUSR has *USE	
ICPGMLIB	OWNIC	*EXCLUDE		
Files available for Query in ITEMLIB	OWNIC	*USE		
Files not available for Query in ITEMLIB	OWNIC	*EXCLUDE		
Programs in ICPGMLIB	OWNIC	*USE		

Note: A special owner profile can be created for objects used by multiple applications.

When USERA selects option 1 (Inventory Control) from MENU1, program ICSTART runs. The program adopts the authority of OWNIC, giving *ALL authority to the inventory control objects in ITEMLIB and the programs in ICPGMLIB. USERA is thus authorized to make changes to the inventory control files while using options from the ICMENU.

When USERA exits ICMENU and returns to MENU1, the ITEMLIB and ICPGMLIB libraries are removed from the USERA library list, and program ICSTART is removed from the call 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 194 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 that 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.

Related concepts

Planning file security

The information contained in database files is often the most important asset on your system. Resource security allows you to control who can view, change, and delete information in a file.

Ignoring adopted authority

The technique of using adopted authority in menu design 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.

Figure 39 on page 234 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 39. Sample application menu with query

The authority information for the QRYSTART program is the same as shown in Table 127 on page 233. 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.

Here are comparisons of the call stacks when USERA selects query from MENU1 (see Figure 36 on page 232) and from ICMENU:

Call stack when query selected from MENU1

- MENU1 (no adopted authority)
- QRYSTART (adopted authority QRYUSR)

Call 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 a query from ICMENU without having the ability to change and delete files. This is because the authority of OWNIC is not used by the QRYSTART program.

When USERA ends the 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 34 on page 231) 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 call stack.

Related concepts

Programs that ignore adopted authority
You can specify the use adopted authority (USEADPAUT) parameter to control whether a program uses the adopted authority.

Related reference

Flowchart 8: How adopted authority is checked
If insufficient authority is found by checking user authority, the system checks adopted authority.

Related information

Scenarios for HTTP Server

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. This might represent a security exposure because the public authority to the System Request Menu is *USE when a system is shipped.

The simplest way to prevent users from accessing this menu is to restrict 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)
```

Some of the actual commands used for the System Request menu come from the CPX2313 message in the QCPFMSG message file. Commands are qualified with a library name from the CPX2373 message. The values in the CPX2373 message for each command are *NLVLIBL or *SYSTEM. Someone might potentially use the Override Message File (OVRMSGF) command to change the commands that the System Request menu options use.

Each time the System Request key is pressed, the system automatically changes the current user profile of the job to the initial user profile of the job. This is done so that the user does not have any additional authority on the System Request menu or in the Presystem Request Program exit program. After the

System Request function is completed, the current user profile of the job is returned to the value that it was before the System Request key was pressed.

You can prevent users from selecting specific options from the System Request Menu by restricting the authority to the associated commands. Table 128 on page 236 shows the commands associated with the menu options:

Table 128. 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 Workstation 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.)
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. 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.

Related concepts

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.

Planning command security

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 might 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 that command. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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 might restrict access to the entire library. If you use one or more national language versions of the IBM i 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 often 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.

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.

Related reference

Using adopted authority in menu design

The availability of decision-support tools, such as Query/400, poses challenges for security design. 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.

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.

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 40 on page 238 shows an example of a physical file and three different logical files associated with it.

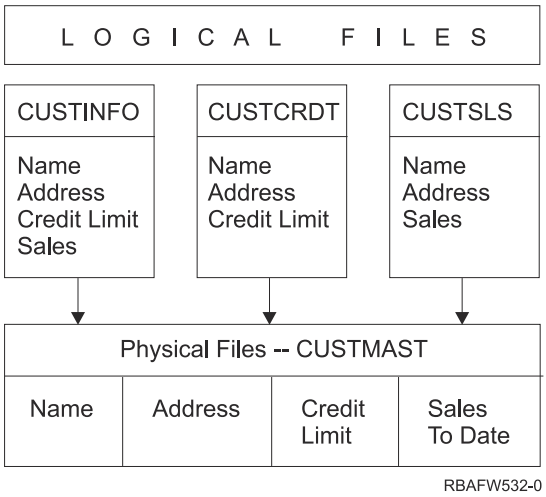

Figure 40. Using a logical file for security

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 129. Physical file example: CUSTMAST file

Authority	**Users: *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 authority but no object operational authority 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 authority 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 190 shows how using primary group authority affects the authority checking process.

You can specify data authorities for logical files beginning with V3R1 of the IBM i licensed program. When a pre-V3R1 logical file is restored on a V3R1 system or later, the system converts your logical files 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.

Related information

Db2 for i

Overriding files

You can use override commands 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 can 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 that a program processes the correct files by using override commands with SECURE(*YES) before the program runs, thus those files are protected from the effects of any file override commands that were previously called. If you use SECURE(*NO), those files are not protected from other file overrides. Their values can be overridden by the effects of any file override commands that were previously called.

File security and SQL

Relational databases use catalog tables and views to store information about all database objects, their attributes, privileges, connection information to other relational databases, and much more. The catalog tables and views are heavily used by client interfaces such as JDBC, ODBC, .NET, and CLI. They are also necessary to support certain Structured Query Language (SQL) statements. Many user, third party, and IBM i applications also depend on direct access to the catalog tables and views. For example, IBM Navigator for i heavily uses the catalog tables and views. Since many types of applications depend on the catalog tables and views by default, they are generally granted public authority of SELECT (*OBJOPR and *READ). Applications that use only native database access do not implicitly use the catalog tables and views.

Planning group profiles

A group profile is a useful tool when several users have similar security requirements. You can directly create group files or you can make an existing profile into a group profile. When you use group profiles, you can manage authority more efficiently and reduce the number of individual private authorities for objects.

Group files 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.

A group profile is just a special type of user profile. It becomes a group profile when one of the following conditions are met:

- 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.

Related concepts
Group profiles

A *group profile* is a special type of user profile. Rather than giving authority to each user individually, you can use a group profile to define authority for a group of users.

Considerations for 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 about 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.

Related tasks

Case 2: Using primary group authority
This case demonstrates how to use primary group authority.

Considerations for multiple group profiles

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 effect on the performance of authority checking. This topic provides some suggestions on using 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).

Here are 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 172. Make sure that you understand how using multiple groups in combination with other authority techniques, such as authorization lists, might affect your system performance.

Accumulating special authorities for group profile members

Special authorities are cumulative for users who are members of multiple groups.

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: If a group member owns a program, the program adopts only the authority of the owner. The authorities of the owner's 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 might find that a specific user has all of 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 might 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 actions:

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 130 on page 242 shows the characteristics of the two methods.

Table 130. 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	Yes [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. Private group authorities are saved if PVTAUT(*YES) is specified on the save command.

For the authorization list of the item "Authority can be specified when the object is created":

- To assign an authorization list to a library-based object, specify AUT (*LIBCRTAUT) on the CRTxxxx command and the CRTAUT (authorization-list-name) for the library. Some objects, such as validation lists, cannot use a value of *LIBCRTAUT in the CRT command.
- To assign an authorization list to a directory-based object, specify the *INDIR value for the DTAAUT and OBJAUT parameters on the MKDIR command. In this way, the authorization list secures both the parent directory and the new one. The system does not allow an arbitrary authorization list to be specified when an object is created.

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.

Programmers 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. If you must give programmers special authorities, give them only the special authority that is required to perform the jobs or tasks that are 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 that are 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 147 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 that is designated for the application.

  Application objects should not be owned by a programmer because the programmer can have uncontrolled access to them in a production environment. The profile that owns the application might be the profile of the individual responsible for the application, or it might be a profile specifically created as the application owner.

Managing source files

To protect the information on your system, you need carefully plan the security of source files.

Source files are important to the integrity of your system. They might 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 who can move them to production.

When a source file is created on the system, the default public authority is *CHANGE. This 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 in order to add new members. The public authority should be reduced to *USE or *EXCLUDE, unless the source files are in a controlled library.

Protecting Java class files and jar files in the integrated file system

To run a Java program, you will need read (*R) authority to each Java class and jar file plus execute (*X) authority to each directory in the path to the Java class and jar files. If you use Java class and jar files in the integrated file system, you need to protect them using normal object authorities.

To protect Java files, use the **CHGAUT** command to secure the directories in the path and the files with object authority attributes. A user might need read (*R) authority to the Java class and jar files to run a Java program. They can get that authority from the public authority of the file or from private authority. An authorization list is helpful in setting up private authority for a group of users. Do not give anyone write (*W) authority to the file unless they are allowed to change the file.

You can use the Classpath Security Check Level (CHKPATH) parameter on the **RUNJVA** command to make sure that a running Java application is using the correct files from the CLASSPATH. You can use a value of CHKPATH(*SECURE) to prevent a Java program from running if one or more warning messages are sent for each directory in the CLASSPATH that has public write authority.

Planning security for system programmers or managers

You can limit the authority given to system programmers or managers to protect the files on your system.

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.

For example, you might want Sue and Fred to be the two people who can create and change user profiles without giving them special authorities. You can achieve this by doing the following steps.

1. Write a command or program that is a front end to the **CRT/CHGUSRPRF** command.
2. Have the command or program adopt a profile that can do the creates and changes.
3. Authorize Sue and Fred to the program.

Then Sue and Fred can only do the task through the application.

Mitigating Spectre and Meltdown vulnerabilities in new and existing programs

Determine whether programs should be mitigated for Spectre and Meltdown vulnerabilities. For more information see Mitigating Spectre and Meltdown vulnerabilities in new and existing programs in the Planning and setting up system security topic.

Using validation lists

Validation list objects provide a method for applications to securely store user-authentication information.

For example, the Internet Connection Server (ICS) uses validation lists to carry out the concept of an Internet user. 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 a IBM i user id and password.

An internet user can be permitted or denied access to the system from the Web server. The user, however, has no authority to any IBM i resources or authority to sign-on or run jobs. A IBM i 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.

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. Since the validation list is encrypted, this method is more secure than using the application alone to verify the user's password.

You can 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.

Related reference

Retain Server Security (QRETSVRSEC)
The Retain Server Security (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 IBM i user profile password.

Related information

Application programming interfaces

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 function goes through the following processes to do the verification.

- 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

The limit access to program function lets APIs perform the following tasks: To use this function 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, before the application invokes the code block, it calls the check usage API to verify that the user has the authority to use the function that is associated with the code block. If the user is allowed to use the registered function, the code block runs. 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 Work with Function Usage Information (**WRKFCNUSG**) command to manage the access to program function or use Application Administration in the IBM Navigator for i.

Related information

Application administration

Separation of duties

Separation of duties helps businesses comply with government regulations and simplifies the management of authorities. It provides the ability for administrative functions to be divided across

individuals without overlapping responsibilities, so that one user does not possess unlimited authority, such as with *ALLOBJ authority. The function, QIBM_DB_SECADM, provides a user with the ability to grant authority, revoke authority, change ownership, or change primary group, but without giving access to the object or, in the case of a database table, to the data that is in the table or allowing other operations on the table.

QIBM_DB_SECADM function usage can be given only by a user with *SECADM special authority and can be given to a user or a group.

QIBM_DB_SECADM is also responsible for administering Row and Column Access Control. Row and Column Access Control provides the ability to restrict which rows a user is allowed to access in a table and whether a user is allowed to see information in certain columns of a table. For more information, see Row and column access control (RCAC)

Chapter 8. Backup and recovery of security information

Saving your security information is just as important as saving your data. In some situations, you might need to recover user profiles, object authorities, and the data on your system. If you do not have your security information saved, you might need to manually rebuild user profiles and object authorities. This can be time-consuming and can lead to errors and security exposures.

This topic includes information on the following topics:

- How security information is saved and restored
- How security affects saving and restoring objects
- Security issues associated with *SAVSYS special authority

Planning adequate backup and recovery procedures for security information requires understanding how the information is stored, saved, and restored.

Table 131 on page 247 shows the commands that are used to save and restore security information. The sections that follow discuss saving and restoring security information in more detail.

Table 131. How security information is saved and restored

Security information saved or restored	Save and restore commands used					
	SAVSECDTA SAVSYS	**SAVCHGOBJ SAVOBJ SAVLIB SAVDLO SAVCFG**	**RSTUSRPRF**	**RSTOBJ RSTLIB RSTDLO RSTCFG**	**RSTAUT**	**RSTDFROBJ**
User profiles	X		X			
Object ownership [1]		X		X		X
Primary group [1]		X		X		X
Public authorities [1]		X		X		X
Private authorities[3]	X	X	X	X	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	
Validation lists		X		X		
Server Authentication Entries	X		X			

Table 131. How security information is saved and restored (continued)

Security information saved or restored	Save and restore commands used					
	SAVSECDTA SAVSYS	**SAVCHGOBJ SAVOBJ SAVLIB SAVDLO SAVCFG**	**RSTUSRPRF**	**RSTOBJ RSTLIB RSTDLO RSTCFG**	**RSTAUT**	**RSTDFROBJ**

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.

3

Private authorities for all objects are saved with **SAVSECDTA**. **RSTUSRPRF** will restore the authority information needed to restore the private authorities. The private authorities are restored with **RSTAUT**. Private authorities for individual objects can be saved with the **SAV**, **SAVLIB**, **SAVOBJ**, and **SAVCHGOBJ** commands. Private authorities for individual objects can be restored with the **RST**, **RSTLIB**, and **RSTOBJ** commands if they were saved with the save command.

Related information

Backup and recovery

How security information is stored

Planning adequate backup and recovery procedures for security information requires understanding how the information is stored and saved.

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.
- *Server Authentication Information*:
  - Server authentication entries.

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.

Related concepts

Additional information associated with a user profile
This topic discusses the private authorities, owned object information, and primary group object information that are associated with a user profile.

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
- Private authorities for the object, if PVTAUT(*YES) is specified on the SAVxxx command

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.
- Other application information associated with the user profile. For example:
  - Server authentication entries

– User Application Information entries that are added using the Update User Application Information (QsyUpdateUserApplicationInfo) API

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 typical sequence for recovery is:

1. Restore user profiles and authorization lists (RSTUSRPRF USRPRF(*ALL)).
2. Restore objects (RSTCFG, RSTLIB, RSTOBJ, RSTDLO or RST).
3. Restore the private authorities to objects (RSTAUT).

Note: By using the code examples, you agree to the terms of the Chapter 11, “Code license and disclaimer information,” on page 333.

Related information

Recovering your system

Restoring user profiles

There might be some changes that are made to a user profile when it is restored.

The following rules apply:

- If profiles are being restored individually (RSTUSRPRF USRPRF(*ALL) is not specified), SECDTA(*PWDGRP) is not requested, and the profile that is 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 of the user profiles are being restored to your system, all of the fields in any of the profiles that already exist on the system are restored from the save media, including the password.

  Attention:

1. User Profiles saved from a system with a different password level (QPWDLVL system value) than the system that is being restored might 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 can have a password of "This is my password". This password will not be valid on a system running password level 0 or 1.
2. Keep a record of the security officer (QSECOFR) password associated with each version of your security information that is saved. This ensures that 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.

- 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 that are 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 a 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 to 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

- If a profile is restored (all profiles or individual profile) that already exists on the system, the restore operation will not change the existing user expiration fields.
- If a profile is restored (all profiles or individual profile) that does not yet exist on the system, all fields in the user profile are restored from the save media, including the user expiration interval and user expiration date fields:
  - If the profile is enabled and user expiration date is past, the user profile will be set to disabled and CPF2271 diagnostic message will be sent.
  - If the profile is enabled and the user expiration date has not past, the job scheduler entry will be added.

Related information

Resetting the QSECOFR IBM i user profile password

Restoring objects

When you restore an object to the system, the system uses the authority information stored with the object. This topic describes the rules applicable to the authority information when restoring objects.

The following applies to the security of the restored object:

Object ownership:

- If the profile that owns the object exists 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), ALWOBJDIF(*OWNER), or ALWOBJDIF(*COMPATIBLE) is specified. In that case, the object is restored and the owner on the system is used.
- See "Restoring programs" on page 254 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 that is being restored does not exist on the system, public authority is set to the public authority of the saved object.
- If the object that is 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), ALWOBJDIF(*AUTL), or ALWOBJDIF(*COMPATIBLE) is specified, the object is restored. The object is linked to the authorization list that is associated with the existing object.
- If a document or folder that already exists on the system is restored, the authorization list that is 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), ALWOBJDIF(*AUTL), or ALWOBJDIF(*COMPATIBLE) 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, and with objects if PVTAUT(*YES) is specified on the SAVxxx command.
- If user profiles have private authority to an object that is being restored, those private authorities are typically not affected. Restoring certain types of programs might result in private authorities being revoked.

- If an object is deleted from the system, the 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 the object is then restored from a save version, the private authorities can be restored if PVTAUT(*YES) was specified when the object was saved.
- If private authorities need to be recovered and the private authorities were not saved with the object, then 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 that is being restored does not exist on the system, the object auditing (OBJAUD) value of the saved object is restored.
- If the object that is 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 or directory that is being restored does not exist on the system, the create object or directory auditing (CRTOBJAUD) value for the library or directory is restored.
- If a library or directory that is being restored exists and is being replaced, the CRTOBJAUD value for the library or directory is not restored. The CRTOBJAUD value for the existing library or directory is used.

Authority holder:

- If a file is restored and an authority holder exists for that file name as well as 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:

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.

Related concepts

Restoring programs

Restoring programs to your system that are obtained from an unknown source poses a security exposure. This topic provides information about the factors that should be taken into consideration when restoring programs.

Restoring authorization lists

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.

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 by using the information from the authority table. The grant authority operation runs for each private authority in the authority table. This can be a lengthy process if authority is being restored for many profiles and if many private authorities exist in the authority tables.

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 that was created by the **SAVSECDTA** command, the SAVSYS command, or the QSRSAVO API to find the profiles you want to restore.

Note: Do not run the Reclaim Storage (RCLSTG) command between the Restore User Profile (RSTUSRPRF) command and the RSTAUT command. The RCLSTG command deletes authority reference tables.

If the private authorities are saved with objects, you can optionally restore them with the objects. This is suggested if you are saving and restoring a relatively small number of objects, rather than an entire system.

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. This topic provides information about the factors that should be taken into consideration when restoring programs.

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.

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 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.

Programs converted at restore time can be mitigated for Spectre and Meltdown vulnerabilities, if desired. For more information see Mitigating Spectre and Meltdown vulnerabilities in new and existing programs in the Planning and setting up system security topic.

Restoring programs that adopt the owner's authority:

When a program that adopts owner authority is restored, the ownership and authority to the program might 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.

Related concepts

Restoring objects

When you restore an object to the system, the system uses the authority information stored with the object. This topic describes the rules applicable to the authority information when restoring objects.

Related reference

Security-related restore system values

This topic introduces the security-related restore system values on your IBM i operating system.

Restoring licensed programs

This topic introduces the instructions on restoring the licensed programs on your system.

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 that were created by using the IBM System Manager for IBM i 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. Instead of having a user with *ALLOBJ authority run the command directly, use a program that adopts *ALLOBJ special authority to run the **RSTLICPGM** command.

Here 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. In this example, the user profile is called OWNCP.
2. Write a program to install the application. In this example, the program is called CPINST:

   Note: By using the code examples, you agree to the terms of the Chapter 11, “Code license and disclaimer information,” on page 333.

   ```
   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

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. Users' private authorities to the list are restored using the **RSTAUT** command.

Authorization lists are saved by either the **SAVSECDTA** command or the **SAVSYS** command. Authorization lists are restored by the command:

```
RSTUSRPRF USRPRF(*ALL)
```

Recovering from a damaged authorization list

When an authorization list that secures an object 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.

Related concepts

Restoring objects

When you restore an object to the system, the system uses the authority information stored with the object. This topic describes the rules applicable to the authority information when restoring objects.

Recovering the authorization list

Use the instructions in this topic to recover the authorization list.

If users’ authorities to the authorization list are known, you can restore the authorization list by following the steps below.

1. Delete the authorization list.
2. Create the authorization list again.
3. Add all known users to it.

If you do not know all of the user authorities, you can restore the authorization list by using the last saved SAVSYS or SAVECDTA tapes. To restore the authorization list, do the following actions:

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 by using the RSTAUT command.

This procedure restores user profile values from the save media. Refer to “Restoring user profiles” on page 250 for more information about restoring user profiles values from save media.

Recovering the association of objects to the authorization list

Follow the steps in this topic to recover the association of objects to the authorization list.

When the damaged authorization list is deleted, the objects that were secured by the authorization list need to be added to the new authorization list. Do the following actions:

1. Find the objects that were associated with the damaged authorization list by 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 that are 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)
```

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 actions:

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.

Notes:

1. If you no longer want to secure the installation of the operating system, follow the same steps and specify 2 (not secure).

*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 might 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 IBM i

This section 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 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 section are appropriate for all of these situations. Which things you audit and how often depends on the size and security needs of your organization. The purpose of this section 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 section 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 about the system.

Security auditing involves using commands in the IBM i environment and accessing log and journal information about the system. You might 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 section require a user profile with *ALLOBJ and *SECADM special authority. Make sure that you set the password for the auditor profile to *NONE when the audit period has ended.

Related concepts

Security audit journal

You can use security audit journals to audit the effectiveness of security on your system.

Checklist for security officers and auditors

You can use the checklist to plan and audit your system's security.

As you plan security, choose the subjects from this collection that best meet your security requirements. When you audit the security of your system, use the list to evaluate the controls that you have in place and to determine if additional controls are needed.

Each list serves as a review of the information in this topic collection. They contain brief descriptions of how to do each item and how to verify that the item has been done, including what entries in the QAUDJRN journal to look for. Details about the items are found throughout this topic collection.

Physical security

You can use the physical security checklist to plan or audit physical security of your system.

Note: See Planning and setting up system security for a complete discussion of physical security on the IBM i product.

Here is a checklist for planning physical security of your system:

__ • The system unit and console are in a secure location.

__ • Backup media is protected from damage and theft.

__ • 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.

__ • Consider the physical location for printers, tape devices, fax machines, networking equipment, etc. to ensure that they are in a secure location. Sensitive data often is printed or sent by fax. Tape, or other removable media, contains data that needs to be secured. Networking equipment should be physically secured to ensure it cannot be disconnected or configuration settings changed (ports opened or closed, etc.).

__ • Consider using hardware that encrypts backup media (tape encryption) and consider using encryption capable disk hardware to encrypt the data that is written to disk drives. Encrypting data on tape protects data in the event the physical media (tape) is lost or stolen. Encrypting data on disk will protect data in the event of a disk drive failure and you lose physical control of the broken disk drive after it has been removed or replaced.

System values

Setting up the auditing function for system values helps you to track the changed values on the system.

- 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.
- Use the Display Security Attributes (DSPSECA) command to verify the current and pending values of QSECURITY (security level) and QPWDLVL (password level), and the current setting of the security related system (whether the values can be changed).
- Review decisions about system values periodically. This is particularly important when the system environment changes, such as the installation of new applications or a communications network.

IBM-supplied user profiles

You can perform auditing tasks on IBM-supplied user profiles by verifying their passwords.

- 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 to 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.
- The IBM passwords for dedicated service tools (DST) are changed.

User IDs for service tools do not appear on a DSPAUTUSR list. To verify that the user IDs and passwords are changed, start DST and attempt to use the default values.

- With the exception of QSECOFR, do not sign on with the IBM-supplied user profiles.

  These IBM-supplied profiles are designed to own objects or to run system functions. Use a DSPAUTUSR list to verify that the IBM-supplied user profiles listed in Appendix B, "IBM-supplied user profiles," on page 345, except QSECOFR, have a password of *NONE.

Related concepts

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.

Working with service tools user IDs
You can manage service tools user IDs using system service tools (SST), dedicated service tools (DST), and CL commands.

Related reference

IBM-supplied user profiles
This section 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.

Password control

You can use the password control mechanism to audit your system security.

- 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, such as 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 attempts to create an equivalent password that is usable at the other password levels. You can use the PRTUSRPRF TYPE(*PWDLVL) command 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 can 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 not case sensitive at password levels 0 and 1).

User and group profiles

You can validate the user and group profiles and their authorities to audit the security effectiveness on your system.

- Each user is assigned a unique user profile.

  Set the QLMTDEVSSN system value 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.

  Use the DSPUSRPRF command to check the special authorities for user profiles and to determine which profiles are group profiles. The topic "Printing selected user profiles" on page 311 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 311 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 243 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 311 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. To make sure user profiles are deleted immediately after employees leave, review the DO (Delete Object) entries in the audit journal.

- Management regularly verifies the users authorized to the system.

  Use the DSPAUTUSR command to view users authorization 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 311 gives an example of how to determine this.

Authorization control

Authorization control enables you to audit the security of the information stored on your system.

You can use the following checklist to help you audit authorization control security.

- 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 Display Object Authority (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 the 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 208 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 312 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

Use this checklist along with auditing journal to audit unauthorized attempts to access information.

- 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 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 make sure that no subsystem descriptions have a workstation entry that uses a job description that has a user profile name.

Unauthorized programs

The Check Object Integrity (CHKOBJITG) command allows you to audit unauthorized changes to program changes on the system.

- 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 313.

Communications

This checklist can be used to plan and audit the controls needed over various types of communications on the system.

- Use call-back procedures to protect telephone communications.
- Use encryption on sensitive data.
- Control remote sign-on. The QRMTSIGN system value is set to *FRCSIGNON or a pass-through validation program is used.
- Use the JOBACN, PCSACC, and DDMACC network attributes to control access to data from other systems, including personal computers. The JOBACN network attribute should be *FILE.

Using the security audit journal

The security audit journal is the primary source of auditing information about the system. This section describes how to plan, set up, and manage security auditing, what information is recorded, and how to view that information.

A security auditor inside or outside your organization can use the auditing function that is 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 265 describes how to do this.

When a security-related event that might 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. You can use an application program or a query tool to analyze the data.

Related reference

Layout of audit journal entries
This section 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.

Object operations and auditing
This topic collection lists operations that can be performed against objects on the system, and whether those operations are audited.

Planning security auditing

The security auditing function is optional. You must take specific steps to set up security auditing.

To plan the use of security auditing on your system, follow these steps:

- 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, the QAUDLVL2 (audit level extension) system value, and the AUDLVL (action auditing) parameter in user profiles work together to control action auditing.

The functions of each system value are as follows:

- The QAUDLVL system value specifies which actions are audited for all users of the system.
- The QAUDLVL2 system value also specifies which actions are audited for all users of the system, and is used when more than 16 auditing values are needed.
- 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 and QAUDLVL2 system values.
- The QAUDCTL system value starts and stops action auditing.

The events that you choose to log depends on both your security objectives and your potential exposures. “Action auditing” on page 118 describes the possible audit level values and how you can use them. It shows whether they are available as a system value, a user profile parameter, or both.

Related reference

Auditing Level (QAUDLVL)
The Auditing Level (QAUDLVL) system value along with the QAUDLVL2 system value determines which security-related events are logged to the security audit journal (QAUDJRN) for all system users.

Auditing Level Extension (QAUDLVL2)
The Auditing Level Extension (QAUDLVL2) system value is required when more than sixteen auditing values are needed.

Action auditing

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 and QAUDLVL2 system values.

Action auditing values

This table lists the possible values available on the QAUDLVL and QAUDLVL2 system values and the CHGUSRAUD command when auditing actions of the system.

Table 132. Action auditing values

Possible value	Available on QAUDLVL and QAUDLVL2 system values	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.
*ATNEVT	Yes	No	**Attention events**: The system writes a journal entry for events that require further examination. With this information, you can determine the potential significance of the attention event to the system.
*AUTFAIL	Yes	Yes	**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), then only the command name and library name are logged. *CMD can 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 can 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.
*JOBBAS	Yes	Yes	**Job base functions**: Actions that affect a job are logged, such as starting or stopping a job, holding, releasing, canceling, or changing the job.

Table 132. Action auditing values (continued)

Possible value	Available on QAUDLVL and QAUDLVL2 system values	Available on CHGUSRAUD command	Description
*JOBCHGUSR	Yes	Yes	**Job change user**: Changes to a thread's active user profile or its group profiles are logged.
*JOBDTA	Yes	Yes	**Job tasks**: Actions that affect a job are logged, such as starting or stopping a job, holding, releasing, canceling, or changing the job, changing the thread's active user profile or group profile. *JOBDTA can be used to monitor who is running batch jobs. *JOBDTA is composed of two values, which are *JOBBAS and *JOBCHGUSR, to enable you to better customize your auditing.
*NETBAS	Yes	Yes	**Network base functions**: IP rules actions, sockets connections, APPN directory search filter, APPN end point filter.
*NETCLU	Yes	Yes	**Cluster or cluster resource group operations**: An audit journal entry is written when any of these events occur: • A cluster node or cluster resource group is added, created, or deleted. • A cluster node or cluster resource group is started, ended, updated, or removed. • Automatic failure of a system that switches access to another system. • Access is manually switched from one system to another system in a cluster.
*NETCMN	Yes	Yes	**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. *NETCMN is composed of several values to allow you to better customize your auditing. The following values make up *NETCMN: *NETBAS *NETCLU *NETFAIL The Mail and DHCP functions from *NETSCK
*NETFAIL	Yes	Yes	**Network failures**: An audit journal entry is written when trying to connect to a TCP/IP port that does not exist, or trying to send information to a TCP/IP port that is not open or available.

Table 132. Action auditing values (continued)

Possible value	Available on QAUDLVL and QAUDLVL2 system values	Available on CHGUSRAUD command	Description
*NETSCK	Yes	Yes	**Socket tasks**: An audit journal entry is written when any of these events occur: • An inbound TCP/IP socket connection is accepted. • An outbound TCP/IP socket connection is established. • An IP address is assigned through DHCP (Dynamic Host Configuration Protocol). • An IP address is unable to be assigned through DHCP because all of the IP addresses are being used. • Mail is filtered or rejected.
*NETSECURE	Yes	Yes	**Secure Connections**: An audit journal entry is written when a secure connection is established.
*NETTELSVR	Yes	No	**Telnet Server Connections**: An audit journal entry is written when a Telnet server connection is accepted.
*NETUDP	Yes	Yes	**User Datagram Protocol (UDP)**: An audit journal entry is written for UDP inbound and outbound traffic.
*OBJMGT	Yes	Yes	**Object management tasks**: Moving an object to a different library or renaming it is logged. *OBJMGT can 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 can be used to detect attempts to create or delete an optical directory.
*PGMADP	Yes	Yes	**Adopting authority**: The system writes a journal entry when adopted authority is used to gain access to an object. *PGMADP can be used to test where and how a new application uses adopted authority.
*PGMFAIL	Yes	Yes	**Program failures**: The system writes a journal entry when a program causes an integrity error. *PGMFAIL can be used to assist with migration to a higher security level or to test a new application.

Table 132. Action auditing values (continued)

Possible value	Available on QAUDLVL and QAUDLVL2 system values	Available on CHGUSRAUD command	Description
*PRTDTA	Yes	Yes	**Printing functions**: Printing a spooled file, printing directly from a program, or sending a spooled file to a remote printer is logged. *PRTDTA can be used to detect printing confidential information.
*PTFOBJ	Yes	No	**Program Temporary Fix (PTF) objects**: Changes to PTF objects during PTF operations are audited. The objects include library objects such as *PGM and *SRVPGM objects, replaceable Unit (RU) objects for LIC PTFs, and Integrated File System (IFS) objects.
*PTFOPR	Yes	No	**Program Temporary Fix (PTF) operations**: An audit record is written when any of these operations occur: • Load, apply, or remove of a PTF. • Log or delete of a PTF save file. • Install of a PTF using the GO PTF or INSPTF command.
*SAVRST	Yes	Yes	**Restore operations**: *SAVRST can be used to detect attempts to restore unauthorized objects.
*SECCFG	Yes	Yes	**Security configuration**: An audit journal entry is written when any of these events occur: • User profiles are created, changed, deleted, or restored. • Changes are made to programs, system values, subsystem routing, or to the auditing attributes of an object. • The QSECOFR password is reset to the shipped value. • The service tools security officer password is defaulted.
*SECDIRSRV	Yes	Yes	**Directory service functions**: An audit journal entry is written when any of these events occur: • Changes or updates are made to auditing, authority, passwords, and ownership. • Successful binds and unbinds. • Changes are made to directory security policies (for example, password policy)

Table 132. Action auditing values (continued)

Possible value	Available on QAUDLVL and QAUDLVL2 system values	Available on CHGUSRAUD command	Description
*SECIPC	Yes	Yes	**Interprocess communications**: An audit journal entry is written when any of these events occur: • Changes are made to the ownership or authority of an IPC object. • A create, delete, or retrieve of an IPC object. • Shared memory attach.
*SECNAS	Yes	Yes	**Network authentication service actions**: An audit journal entry is written when any of these events occur: • Service ticket invalid. • Service principals do not match. • Client principals do not match. • Ticket IP address mismatch. • Decryption of the ticket failed. • Decryption of the authentication failed. • Realm is not within client and local realms. • Ticket is a replay attempt. • Ticket not yet valid. • Remote or local IP address mismatch. • Decryption of KRB_AP_PRIV or KRB_AP_SAFE checksum error. • For KRB_AP_PRIV or KRB_AP_SAFE: Timestamp error, replay error, or sequence order error. • For graphics symbol set accept: Expired credentials, checksum error, or channel bindings. • For graphics symbol set unwrap or graphics symbol set verify: Expired context, decrypt/decode, checksum error, or sequence error.
*SECRUN	Yes	Yes	**Security runtime functions**: Changes to object ownership, authority, and primary group are written to the audit journal.
*SECSCKD	Yes	Yes	**Socket descriptors**: An audit journal entry is written when any of these events occur: • A socket descriptor is given to another job. • A socket descriptor is received. • A socket descriptor is unusable.

Table 132. Action auditing values (continued)

Possible value	Available on QAUDLVL and QAUDLVL2 system values	Available on CHGUSRAUD command	Description
*SECVFY	Yes	Yes	**Verification functions**: An audit journal entry is written when any of these events occur: • A profile handle or token is generated. • All profile tokens were invalidated. • The maximum number of profile tokens has been generated. • All profile tokens for a user have been removed. • A user profile has been authenticated. • A target profile was changed during a pass-through session.
*SECVLDL	Yes	Yes	**Validation list operations**: An audit journal entry is written when any of these events occur: • An add, change, remove, or find of a validation list entry. • Successful or unsuccessful verification of a validation list entry.
*SECURITY	Yes	Yes	**Security tasks**: Security-relevant events, such as changing a user profile or system value, are logged. *SECURITY can be used to keep a record of all security activity. *SECURITY is composed of several values to allow you to better customize your auditing. The following values make up *SECURITY: *SECCFG *SECDIRSRV *SECIPC *SECNAS *SECRUN *SECSCKD *SECVFY *SECVLDL
*SERVICE	Yes	Yes	**Service tasks**: The use of service tools, such as DMPOBJ (Dump Object) and STRCPYSCN (Start Copy Screen), is logged. *SERVICE can 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 can be used to detect attempts to print or send confidential data.

Table 132. Action auditing values (continued)			
Possible value	**Available on QAUDLVL and QAUDLVL2 system values**	**Available on CHGUSRAUD command**	**Description**
*SYSMGT	Yes	Yes	**Systems management tasks**: The system writes a journal entry for systems management activities, such as changing a reply list, changing the power on/off schedule, or Db2® Mirror for i actions. *SYSMGT can be used to detect attempts to use systems management functions to circumvent security controls.

Security auditing journal entries

This topic provides information about the journal entries that are written for the action auditing values specified on the QAUDLVL and QAUDLVL2 system values and in the user profile.

It shows:

- The type of entry written to the QAUDJRN journal.
- The model database output file 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, "Layout of audit journal entries," on page 629.
- 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 133. Security auditing journal entries				
Action or object auditing value	**Journal entry type**	**Model database outfile**	**Detailed entry**	**Description**
Action Auditing:				
*ATNEVT	IM	QASYIMJ5	P	A potential intrusion has been detected. Further evaluation is required to determine if this is an actual intrusion or an expected and permitted action.

Table 133. Security auditing journal entries (continued)

Action or object auditing value	Journal entry type	Model database outfile	Detailed entry	Description
*AUTFAIL	AF	QASYAFJE/J4/J5	A	An attempt was made to access an object or perform an operation to which the user was not authorized.
			B	Restricted instruction
			C	Validation failure
			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.
			H	Scan exit program action.
			I	System Java inheritance not allowed
			J	An attempt was 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.
			K	An attempt was made to perform an operation for which the user did not have the required special authority.
			N	The profile token was not a regenerable profile token.
			O	Optical Object Authority failure
			P	An attempt was made to use a profile handle that is not valid on the QWTSETP API.
			R	Hardware protection error
			S	Default signon attempt.
			T	Not authorized to TCP/IP port.
			U	A user permission request was not valid.
			V	The profile token was not valid for generating new profile token.
			W	The profile token was not valid for exchange.
			X	System violation, see description of AF (Authority Failure) journal entries for details

Table 133. Security auditing journal entries (continued)

Action or object auditing value	Journal entry type	Model database outfile	Detailed entry	Description
			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.
	CV	QASYCVJ4/J5	E	Connection ended abnormally.
			R	Connection rejected.
	DI	QASYDIJ4/J5	AF	Authority failures.
			PW	Password failures.
	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.
			C	CHKPWD failure.
			D	An incorrect service tool user ID was entered.
			E	An incorrect service tool user ID password was entered.
			P	An incorrect password was entered.
			Q	Attempted signon (user authentication) failed because user profile was disabled.
			R	Attempted signon (user authentication) failed because password was expired.
			S	SQL decrypt a password that was not valid.
			U	User name not valid.
			X	Service tools user is disabled.
			Y	Service tools user not valid.
			Z	Service tools password not valid.
	VC	QASYVCJE/J4/J5	R	A connection was rejected because of incorrect password.
	VO	QASYVOJ4/J5	U	Unsuccessful verification of a validation list entry.
	VN	QASYVNJE/J4/J5	R	A network logon was rejected because of expired account, incorrect hours, incorrect user ID, or incorrect password.

Table 133. Security auditing journal entries (continued)

Action or object auditing value	Journal entry type	Model database outfile	Detailed entry	Description
	VP	QASYVPJE/J4/J5	P	An incorrect network password was used.
	X1	QASYX1J5	F	Delegate of identity token failed.
			U	Get user from identity token failed.
	XD	QASYXDJ5	G	Group names (associated with DI entry)
*CMD [1]	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.
			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 [2]	AU	QASYAUJ5	A	Add of an EIM association.
	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 created.
	XD	QASYXDJ5	G	Group names (associated with DI entry)
*DELETE [2]	AU	QASYAUJ5	A	Remove of an EIM association.
	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 deleted.
	LD	QASYLDJE/J4/J5	U	Unlink a directory.
	XD	QASYXDJ5	G	Group names (associated with DI entry)
*JOBBAS	JS	QASYJSJ5	A	The ENDJOBABN command was used.
			B	A job was submitted.
			C	A job was changed.

Table 133. Security auditing journal entries (continued)

Action or object auditing value	Journal entry type	Model database outfile	Detailed entry	Description
			E	A job was ended.
			H	A job was held.
			I	A job was disconnected.
			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.
			U	CHGUSRTRC command.
*JOBCHGUSR	JS	QASYJSJ5	M	Change profile or group profile.
			T	Change profile or group profile using a profile token.
*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	Change 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	Change profile or group profile using a profile token.
			U	CHGUSRTRC command.
	SG	QASYSGJE/J4/J5	A	Asynchronous IBM i signal process.
			P	Asynchronous Private Address Space Environment (PASE) signal processed.
	VC	QASYVCJE/J4/J5	S	A connection was started.

Table 133. Security auditing journal entries (continued)

Action or object auditing value	Journal entry type	Model database outfile	Detailed entry	Description
			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.
*NETBAS	CV	QASYCVJE/J4/J5	C	Connection established.
			E	Connection ended normally.
			R	Rejected connection.
	IR	QASYIRJ4/J5	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).
	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.
*NETCLU	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.
*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.

Table 133. Security auditing journal entries (continued)

Action or object auditing value	Journal entry type	Model database outfile	Detailed entry	Description
	IR	QASYIRJ4/J5	L	IP rules have been loaded from a file.
			N	IP rule 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).
	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	D	DHCP address assigned
			F	Filtered mail
			P	Port unavailable
			R	Reject mail
			U	DHCP address denied
*NETFAIL	SK	QASYSKJ4/J5	P	Port unavailable
*NETSCK [7,9]	SK	QASYSKJ4/J5	A	Accept
			C	Connect
			D	DHCP address assigned
			F	Filtered mail
			R	Reject mail
			U	DHCP address denied

Table 133. Security auditing journal entries (continued)

Action or object auditing value	Journal entry type	Model database outfile	Detailed entry	Description
*NETSECURE [8]	SK	QASYSKJ5	S	Secure connection established. This implies traffic flowing over the connection is now protected by a security protocol known to the system. The system explicitly audits System TLS and IPsec from operating system code responsible for creating the secure connection. IPsec entries for UDP are created using the same frequency as defined for *NETUDP[6].
			X	System TLS secure connection error
*NETTELSVR [9]	SK	QASYSKJ5	A	Telnet Server Accept Note: Telnet clients can be configured to retry the connection attempt after an attempt to establish a session is unsuccessful. These Telnet clients will retry indefinitely until the conditions causing the session to fail are eliminated. This can generate a large number of Telnet server audit journal entries.
*NETUDP [9]	SK	QASYSKJ5	I [6]	User Datagram Protocol (UDP) inbound traffic
			O [6]	UDP outbound traffic
*OBJMGT [2]	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

Table 133. Security auditing journal entries (continued)

Action or object auditing value	Journal entry type	Model database outfile	Detailed entry	Description
			B	Back up 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 call stack.
			E	A program ended that adopts owner authority. The end entry is written when the program leaves the call stack. If the same program occurs more than once in the call 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	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.

Table 133. Security auditing journal entries (continued)

Action or object auditing value	Journal entry type	Model database outfile	Detailed entry	Description
			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	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.
*PTFOBJ	PU	QASYPUJ5	D	Directory PTF object was changed.
			L	Library PTF object was changed.
			S	LIC PTF object was changed.
*PTFOPR	PF	QASYPFJ5	I	PTF IPL operation was performed.
			L	PTF product(s) operation was performed.
			P	PTF operation was performed.
*SAVRST [2]	GR	QASYGRJ5	O	ObjectConnect operations.
	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. [3]
	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.[3]
	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.
	RZ	QASYRZJE/J4/J5	A	The primary group for an object was changed during a restore operation.
			O	Auditing of an object was changed with **CHGOBJAUD** command.

Table 133. Security auditing journal entries (continued)

Action or object auditing value	Journal entry type	Model database outfile	Detailed entry	Description
			U	Auditing for a user was changed with **CHGUSRAUD** command.
*SECCFG	AD	QASYADJE/J4/J5	D	Auditing of a DLO was changed with **CHGDLOAUD** command.
			O	Auditing of an object was changed with **CHGOBJAUD** or **CHGAUD** commands.
			S	The scan attribute was changed using **CHGATR** command or the Qp0lSetAttr API, or when the object was created.
			U	Auditing for a user was changed with **CHGUSRAUD** command.
	AU	QASYAUJ5	E	Enterprise Identity Mapping (EIM) configuration change
	CP	QASYCPJE/J4/J5	A	Create, change, or restore operation of user profile when QSYSRESPA API is used.
	CQ	QASYCQJE/J4/J5	A	A *CRQD object was changed.
	CY	QASYCYJ4/J5	A	Cryptographic Coprocessor Access Control function
			F	Cryptographic Coprocessor Facility Control function
			K	Cryptographic Services Master Key function
			M	Cryptographic Coprocessor Master Key function
	DO	QASYDOJE/J4/J5	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
	DS	QASYDSJE/J4/J5	A	Request to reset DST QSECOFR password to system-supplied default using the CHGDSTPWD command.

Table 133. Security auditing journal entries (continued)

Action or object auditing value	Journal entry type	Model database outfile	Detailed entry	Description
			C	DST profile changed using the QSYCHGDS API.
			D	Delete of a service tools user ID using the DLTSSTUSR command.
			H	Change to a service tools user ID using the CHGSSTUSR command.
			P	Change to a service tools user ID password using the QSYCHGDS API.
			R	Create of a service tools user ID using the CRTSSTUSR command.
			S	Change to the service tools security attributes using the CHGSSTSECA command.
	EV	QASYEVJ4/J5	A	Add.
			C	Change.
			D	Delete.
			I	Initialize environment variable space.
	GR	QASYGRJ4/J5	A	Exit program added
			D	Exit program removed
			F	Function registration operation
			R	Exit program replaced
	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.
	PA	QASYPAJE/J4/J5	A	A program was changed to adopt owner authority.
	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.

Table 133. Security auditing journal entries (continued)

Action or object auditing value	Journal entry type	Model database outfile	Detailed entry	Description
			C	Change to system clock.
			E	Change to option
			F	Change to system-wide journal attribute
	VA	QASYVAJE/J4/J5	S	The access control list was changed successfully.
			F	The change of the access control list failed.
			V	Successful verification 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.
*SECDIRSRV	DI	QASYDIJE/J4/J5	AD	Audit change.
			BN	Successful bind
			CA	Authority change
			CP	Password change
			OW	Ownership change
			PO	Policy change
			UB	Successful unbind
*SECIPC	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.
			M	Shared memory attached.
			Z	Close an IPC object.
*SECNAS	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

Table 133. Security auditing journal entries (continued)

Action or object auditing value	Journal entry type	Model database outfile	Detailed entry	Description
			7	Realm is not within client and 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 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
*SECRUN	AX	QASYAXJ5	M	Column mask created, altered, or dropped.
			P	Row permission created, altered, or dropped.
			T	Table altered.
	CA	QASYCAJE/J4/J5	A	Changes to authorization list or object authority.
	OW	QASYOWJE/J4/J5	A	Object ownership was changed.
	PG	QASYPGJE/J4/J5	A	The primary group for an object was changed.
	X2	None		Query manager profile was changed.

Table 133. Security auditing journal entries (continued)

Action or object auditing value	Journal entry type	Model database outfile	Detailed entry	Description
*SECSCKD	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.
*SECURITY	AD	QASYADJE/J4/J5	D	Auditing of a DLO was changed with **CHGDLOAUD** command.
			O	Auditing of an object was changed with **CHGOBJAUD** or **CHGAUD** commands.
			S	Scan attribute change by **CHGATR** command or Qp01SetAttr API
			U	Auditing for a user was changed with **CHGUSRAUD** command.
			G	Get user from identity token successful
	AU	QASYAUJ5	E	Enterprise Identity Mapping (EIM) configuration change
	AX	QASYAXJ5	M	Column mask created, altered, or dropped.
			P	Row permission created, altered or dropped.
			T	Table altered.
	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 when QSYRESPA API is used
	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	Cryptographic Coprocessor Access Control function
			F	Cryptographic Coprocessor Facility Control function
			K	Cryptographic Services Master Key function

Table 133. Security auditing journal entries (continued)

Action or object auditing value	Journal entry type	Model database outfile	Detailed entry	Description
			M	Cryptographic Coprocessor Master Key function
	DI	QASYDIJ4/J5	AD	Audit change
			BN	Successful bind
			CA	Authority change
			CP	Password change
			OW	Ownership change
			PO	Policy change
			UB	Successful unbind
	DO	QASYDOJE/J4/J5	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
	DS	QASYDSJE/J4/J5	A	Request to reset DST QSECOFR password to system-supplied default using the CHGDSTPWD command.
			C	DST profile changed using the QSYCHGDS API.
			D	Delete of a service tools user ID using the DLTSSTUSR command.
			H	Change to a service tools user ID using the CHGSSTUSR command.
			P	Change to a service tools user ID password using the QSYCHGDS API.
			R	Create of a service tools user ID using the CRTSSTUSR command.
			S	Change to the service tools security attributes using the CHGSSTSECA command.
	EV	QASYEVJ4/J5	A	Add.
			C	Change.

Table 133. Security auditing journal entries (continued)

Action or object auditing value	Journal entry type	Model database outfile	Detailed entry	Description
			D	Delete.
			I	Initialize environment variable space.
	GR	QASYGRJ4/J5	A	Exit program added
			D	Exit 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.
	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.

Table 133. Security auditing journal entries (continued)

Action or object auditing value	Journal entry type	Model database outfile	Detailed entry	Description
			M	The 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.
			T	Telnet QIBM_QTG_DEVINIT exit program profile swap.
			U	Telnet QIBM_QTG_DEVINIT exit program profile override.
			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.
			E	Change to option
			F	Change to system-wide journal attribute
	VA	QASYVAJE/J4/J5	S	The access control list was changed successfully.
			F	The change of the access control list failed.
	VO		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

Table 133. Security auditing journal entries (continued)

Action or object auditing value	Journal entry type	Model database outfile	Detailed entry	Description
			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	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 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
	X1	QASYX1J5	D	Delegate of identity token successful
	X2	None		Query manager profile was changed
*SECVFY	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.

Table 133. Security auditing journal entries (continued)

Action or object auditing value	Journal entry type	Model database outfile	Detailed entry	Description
			M	The 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.
			T	Telnet QIBM_QTG_DEVINIT exit program profile swap.
			U	Telnet QIBM_QTG_DEVINIT exit program profile override.
			V	User profile authenticated.
	X1	QASYX1J5	D	Delegate of identity token successful
			G	Get user from identity token successful
*SECVLDL	VO		V	Successful verification of a validation list entry.
*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.
			S	A spooled file was saved.
			T	A spooled file was restored.
			U	A spooled file was changed.
			V	Only non-security relevant spooled files attributes changed.
*SYSMGT	DI	QASYDIJ4/J5	CF	Configuration changes

Table 133. Security auditing journal entries (continued)

Action or object auditing value	Journal entry type	Model database outfile	Detailed entry	Description
			CI	Create instance
			DI	Delete instance
			RM	Replication management
	SM	QASYSMJE/J4/J5	B	Backup options were changed.
			C	Automatic cleanup options were changed.
			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.
			P	The power on/off schedule was changed.
			S	The system reply list was changed.
			T	The access path recovery times were changed.
	M0	QASYM0J5	A	Db2 Mirror setup tools.
	M6	QASYM6J5	A	Db2 Mirror Communication Services - Add Network Redundancy Group (NRG).
			C	Db2 Mirror Communication Services - Change NRG.
			R	Db2 Mirror Communication Services - Remove NRG.
	M7	QASYM7J5	A	Db2 Mirror Replication Services - Add active replication criteria rule.
			D	Db2 Mirror Replication Services - Duplicate replication criteria rules.
			P	Db2 Mirror Replication Services - Activate pending replication criteria rules.
			R	Db2 Mirror Replication Services - Remove active replication criteria rule.
			S	Db2 Mirror Replication Services - Resynchronization of eligible objects.

Table 133. Security auditing journal entries (continued)

Action or object auditing value	Journal entry type	Model database outfile	Detailed entry	Description
			U	User deferred or deleted entries in the Object Tracking List (OTL) using the SQL QSYS2.CHANGE_RESYNC_ENTRIES procedure.
	M8	QASYM8J5	A	Db2 Mirror Product Services - Add IASP.
			C	Db2 Mirror Product Services - Change mirror.
			F	Db2 Mirror Product Services - Change flight recorder.
			I	Db2 Mirror Product Services - Set default inclusion state.
			O	Db2 Mirror Product Services - Takeover.
			R	Db2 Mirror Product Services - Remove IASP.
			S	Db2 Mirror Product Services - Setup mirror.
			T	Db2 Mirror Product Services - Terminate mirror.
			W	Db2 Mirror Product Services - Swap mirror roles.
	M9	QASYM9J5	C	Db2 Mirror Replication State - Change to the replication state of an ASP.
	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.
Object Auditing:				
*CHANGE	DI	QASYDIJ4/J5	IM	LDAP directory import
			ZC	Object change
	ZC	QASYZCJ4/J5	C	Object changes
			U	Upgrade of open access to an object
	AD	QASYADJEJ4/J5	D	Auditing of an object was changed with **CHGOBJAUD** command.

Table 133. Security auditing journal entries (continued)

Action or object auditing value	Journal entry type	Model database outfile	Detailed entry	Description
			O	Auditing of an object was changed with **CHGOBJAUD** command.
			S	Scan attribute change by **CHGATR** command or Qp01SetAttr API
			U	Auditing for a user was changed with **CHGUSRAUD** command.
	AU	QASYAUJ5	E	Enterprise Identity Mapping (EIM) configuration change
	CA	QASYCAJE/J4/J5	A	Changes to authorization list or object authority.
	OM	QASYOMJE/J4/J5	M	An object was moved to a different library.
			R	An object was renamed.
	OR	QASYORJE/J4/J5	N	A new object was restored to the system.
			E	An object was restored that replaces an existing object.
	OW	QASYOWJE/J4/J5	A	Object ownership was changed.
	PG	QASYPGJE/J4/J5	A	The primary group for an object was changed.
	RA	QASYRAJE/J4/J5	A	The system changed the authority to an object being restored.
	RO	QASYROJE/J4/J5	A	The object owner was changed to QDFTOWN during restore operation.
	RZ	QASYRZJE/J4/J5	A	The primary group for an object was changed during a restore operation.
	GR	QASYGRJ4/J5	F	Function registration operations[5]
	LD	QASYLDJE/J4/J5	L	Link a directory.
			U	Unlink 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.

Table 133. Security auditing journal entries (continued)

Action or object auditing value	Journal entry type	Model database outfile	Detailed entry	Description
			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.
			U	Upgrade of open access to an object.
*ALL [4]	CD	QASYCDJ4/J5	C	Command run
	DI	QASYDIJ4/J5	EX	LDAP directory export
			ZR	Object read
	GR	QASYGRJ4/J5	F	Function registration operations[5]
	LD	QASYLDJE/J4/J5	K	Search a directory.
	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 AUDLVL parameter of a user profile. It is not a value for the QAUDLVL system value.

2 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.

3 See the topic "Restoring objects" on page 251 for information about authority changes which might occur when an object is restored.

4 When *ALL is specified, the entries for both *CHANGE and *ALL are written.

5 When the QUSRSYS/QUSEXRGOBJ *EXITRG object is being audited.

6 UDP traffic for the same local and remote address and port is audited only once every 12 hours by default. Refer to The IPCONFIG macro for details on how to change the default interval.

7 Telnet server connections are not audited as part of *NETSCK. Use *NETTELSVR along with *NETSCK if Telnet server connections should be audited.

8 Telnet server secure connections are not audited as part of *NETSECURE. Use *NETTELSVR along with *NETSECURE if Telnet server secure connections should be audited.

9 To audit all TCP and UDP connections in and out of the system specify *NETSCK, *NETUDP, and *NETTELSVR.

Planning the auditing of object access

The IBM i operating system provides the ability to log accesses to an object in the security audit journal by using system values and the object auditing values for users and objects. 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 134 on page 296 shows how the OBJAUD values for the object and the user profile work together.

Table 134. 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 that are accessing a critical object on the system. You can also use object auditing to keep track of all the object that are accessed by a particular user. Object auditing is a flexible tool that enables 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 might generate many more audit records than you can analyze. This can have a severe effect 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.

Here are some examples of how to use object auditing.

- If certain critical files are used throughout your organization, you can 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 by using the techniques described in "Analyzing audit journal entries with query or a program" on page 305.
- If you are concerned about who is using a particular file, you can collect information about all accesses to the 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 that 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 305.

- To audit all object accesses for a specific user, do the following actions:
  1. Set the OBJAUD value for all objects to *USRPRF using the CHGOBJAUD and CHGAUD commands:

```
                     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 might take many hours to run. Setting up object auditing for all objects on the system often is 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.

Related reference

Object auditing

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.

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 newly created objects.

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 might 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 might 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, the system balances the performance effect against the potential loss of information in a power outage. *SYS is the default choice.

If you set the force level to a low number, you minimize the possibility of losing audit records, but you might notice a negative performance effect. 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 Auditing End Action (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 performs the following tasks 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 turn off 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 events take place:

1. The system shuts down immediately (the equivalent of issuing the PWRDWNSYS *IMMED command).
2. SRC code B900 3D10 is displayed.

Next, you must do the following actions:

1. Start an IPL from the system unit. Make sure that the device specified in the console (QCONSOLE) system value is powered on.
2. To complete the IPL, sign on at the console using a user with *ALLOBJ and *AUDIT special authority.

   The system starts in a restricted state with a message indicating that an auditing error caused the system to stop.
3. The QAUDCTL system value is set to *NONE.
4. 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 reset the QAUDCTL value.

Choosing not to audit QTEMP objects

You can choose to not audit QTEMP objects by specifying the *NOQTEMP value.

The value, *NOQTEMP, can be specified as a value for the system value QAUDCTL. If you use the *NOQTEMP value, you must also specify either *OBJAUD or *AUDLVL for the QAUDCTL. 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).

Note: The create of objects into QTEMP library and the delete of objects from QTEMP library are never audited.

Using CHGSECAUD to set up security auditing

Using the CHGSECAUD command, you can activate system security auditing for actions by ensuring that the security journal exists, setting the QAUDCTL system value to *AUDLVL, and setting the QAUDLVL system value to the default set of values. The default set includes *AUTFAIL, *CREATE, *DELETE, *SECURITY, and *SAVRST action audits.

```
CHGSECAUD QAUDCTL(*AUDLVL) QAUDLVL(*DFTSET)
```

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)

Note:
The CHGSECAUD command creates the journal and journal receiver if it does not exist. The CHGSECAUD then sets the QAUDCTL, QAUDLVL, and QAUDLVL2 system values.

Related reference

Options on the Security Tools menu
You can use the Security Tools (SECTOOLS) menu to simplify the management and control of the security on your system with plenty of options and commands that it provides.

Setting up security auditing

With security auditing, you can collect information about security events in the QAUDJRN journal.

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)

Note:
QSYS/QAUDJRN must exist before QAUDCTL can be changed, otherwise the system auditing function doesn't know the journal name and won't find it.

To set up security auditing, do the following steps. You need *AUDIT special authority to complete these steps.

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')
```

   a) 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.

   b) 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.

      It's very helpful to using this type of naming convention if you choose to have the system manage changing your journal receivers.

   c) 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 that you choose to audit. If you use system change-journal management support, the journal receiver thresholds must be at least 100 000 KB. For more information about journal receiver threshold, refer to Journal management.

   d) Specify *EXCLUDE on the AUT parameter to limit access to the information that is 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 that 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 in the creation of the journal receiver. If you choose this option, you do not need 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. Make sure that they are adequately saved before deleting them from the system.

The Journal management topic provides more information about working with journals and journal receivers.

3. Set the audit level (QAUDLVL) system value or the audit level extension (QAUDLVL2) system value by using the WRKSYSVAL command. The QAUDLVL and QAUDLVL2 system values determine which actions are logged to the audit journal for all users on the system. See "Planning the auditing of actions" on page 265.
4. If necessary, set action auditing for individual users by using the CHGUSRAUD command. See "Planning the auditing of actions" on page 265.
5. If necessary, set object auditing for specific objects by using the CHGOBJAUD, CHGAUD, and CHGDLOAUD commands. See "Planning the auditing of object access" on page 296.
6. If necessary, set object auditing for specific users by 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 298.
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 298.
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 the auditing does not start.

Managing the audit journal and journal receivers

The system provides a mechanism for managing the audit journal and journal receivers. You can use the methods described in this topic to audit the security on your system.

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.

The system uses special locking protection to make sure that it 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
- Moving the journal
- Restoring the journal
- **WRKJRN** command

The information recorded in the security journal entries is described in Appendix F, "Layout of audit journal entries," on page 629. 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 might 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 that was 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 in order 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 can 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 completes. You can monitor for this message so that you can know when to do a save of the detached journal receivers.

Attention: The automatic cleanup function that is provided when using Operational Assistant menus does not clean up the QAUDJRN receivers. To avoid problems with disk space, regularly detach, save, and delete QAUDJRN receivers.

See the Journal management topic for complete information about managing journals and journal receivers.

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.

Related information

Journal management

Saving and deleting audit journal receivers

You should regularly detach the current audit journal receiver and attach a new one.

Overview:

Purpose:
Attach a new audit journal receiver; 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)

Note:
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 and take valuable space on auxiliary storage.

It is suggested to have the system manage receivers automatically. 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 might need to specify a large threshold value for the journal receiver. If you are managing receivers manually, you might need to change journal receivers several times a day. If you log only a few events, you might 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

You can follow the steps described in this topic to save or delete the 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 corresponding status.
3. Use the SAVOBJ command to save each receiver. Do not save the currently attached receiver.
4. Use the DLTJRNRCV command to delete each receiver after it is saved.

An alternative to the preceding procedure can be done by using the journal message queue and monitoring for the CPF7020 message which indicates that the system change journal has completed successfully.

Related information

Recovering your system

User-managed journal receivers

You can follow the steps described here to detach, save, or delete journal receivers manually.

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 will receive a warning message.

Stopping the audit function

You might 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 actions:

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 after you change QAUDCTL to *NONE. If you plan to resume security auditing in the future, you should leave the QAUDJRN journal on the system.

   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 security auditing is active. You need to delete these journal receivers. Saving them before deleting them is not necessary, because they do not contain any audit entries.

Analyzing audit journal entries

After you have set up the security auditing function, you can use several different methods to analyze the events that are logged.

- View selected entries at your workstation using the Display Journal (DSPJRN) command.
- Copy selected entries to output files using the Copy Audit Journal Entries (CPYAUDJRNE) or DSPJRN command, and then using a query tool or program to analyze entries.
- Use the Display Audit Journal Entries (DSPAUDJRNE) command.

  Note: IBM has stopped providing enhancements for the DSPAUDJRNE command. The command does not support all security audit record types, and the command does not list all the fields for the records it supports.
- Use the Receive Journal Entry (RCVJRNE) command on the QAUDJRN journal to receive the entries as they are written to the QAUDJRN journal.
- Use SQL to extract details about audit journal entries by using the QSYS2.DISPLAY_JOURNAL() User Defined Table Function (UDTF). For complete details about DISPLAY_JOURNAL(), see DISPLAY_JOURNAL table function.

  This is an example of using DISPLAY_JOURNAL() to find the Change Profile (CP) audit entries that have occurred within the last 24 hours.

```
SELECT journal_code, journal_entry_type, object, object_type, X.*
FROM TABLE (
QSYS2.Display_Journal(
'QSYS', 'QAUDJRN',              -- Journal library and name
JOURNAL_ENTRY_TYPES => 'CP' ,    -- Journal entry types
STARTING_TIMESTAMP => CURRENT TIMESTAMP - 24 HOURS  -- Time window for search
) ) AS x
ORDER BY entry_timestamp DESC;
```

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 actions:

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
Largest sequence number on this screen  . . . . . :00000000000000000012
Type options, press Enter.
  5=Display entire entry


Opt    Sequence  Code  Type  Object      Library     Job       Time
              1   J     PR                          SCPF      10:24:55
              2   T     CA                          SCPF      10:24:55
              3   T     CO                          SCPF      10:24:55
              4   T     CA                          SCPF      10:24:55
              5   T     CO                          SCPF      10:24:55
              6   T     CA                          SCPF      10:24:55
              7   T     CO                          SCPF      10:24:55
              8   T     CA                          SCPF      10:24:56
              9   T     CO                          SCPF      10:24:56
             10   T     CA                          SCPF      10:24:57
             11   T     CO                          SCPF      10:24:57
             12   T     CA                          SCPF      10:24:57
                                                                More...
F3=Exit   F12=Cancel
```

3. Use option 5 (Display entire entry) to see information about a specific entry:

```
                         Display Journal Entry

Object . . . . . :                  Library  . . . . . :
Member . . . . . :
Incomplete data  . . :   No          Minimized entry data : *None
Sequence . . . . . :   1198
Code . . . . . :   T  - Audit trail entry
Type . . . . . :   CO - Create object

            Entry specific data
Column      *...+....1....+....2....+....3....+....4....+....5
00001      'NISAVLDCK  QSYS      *PGM    CLE                  '
00051      '                                                  '
00101      '                                                  '
00151      '                                                  '
00201      '                                                  '
00251      '                                                  '
00301      '                                                  '

                                                                More...
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, “Layout of audit journal entries,” on page 629 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 a query or program, or Run a query or program

Authority:

*USE authority to QSYS/QAUDJRN, *USE authority to journal receiver, and *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 . . . . . . . . .   *TYPE5
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 QADSPJR5 (with record format QJORDJE5) is provided to define these fields when you specify *TYPE5 as the output file format parameter. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630 for more information.

For more information about other records and their output file formats, see Appendix F, "Layout of audit journal entries," on page 629.

If you want to perform a detailed analysis of a particular entry type, use one of the model database outfiles provided. Table 133 on page 272 shows the name of the model database output file for each entry type. Appendix F, "Layout of audit journal entries," on page 629 shows the file layouts for each model database output file.

For example, to create an output file called AUDJRNAF5 in QGPL that includes only authority failure entries:

1. Create an empty output file with the format defined for AF journal entries:

   ```
   CRTDUPOBJ OBJ(QASYAFJ5) FROMLIB(QSYS) +
     OBJTYPE(*FILE) TOLIB(QGPL) NEWOBJ(AUDJRNAF5)
   ```

2. Use the DSPJRN command to write selected journal entries to the output file:

   ```
   DSPJRN JRN(QAUDJRN) … +
      JRNCDE(T) ENTTYP(AF) OUTPUT(*OUTFILE) +
      OUTFILFMT(*TYPE5) OUTFILE(QGPL/AUDJRNAF5)
   ```

3. Use Query or a program to analyze the information in the AUDJRNAF5 file.

Here 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 QASYPWJ5 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 QASYAFJ5 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 QASYAFJ5 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 133 on page 272 shows which journal entry is written for each authority violation message.

Relationship of object Change Date/Time to audit records

Reports written to detect changes to programs, or other objects, are sometimes based on the Change Date/Time field of the object instead of information in the security audit journal. The following list describes reasons why there might be a difference between the date on the object and the date on the source for the object.

- The **CHGPGM** command is used to force program re-creation to update the Change Date/Time field of the program. This operation writes a ZC (Change to Object) audit record.
- The Sign Object (QYDOSGNO) API is used to digitally sign a program or command to update the Change Date/Time field for the program or command. This operation writes a ZC audit record.
- When a device file is opened for update, a ZC audit record is written for the device file. For example, tape device files are opened for update during a save, display device files are opened for update when an application sends and receives data to/from a display device, printer files are opened for update when printed output is produced. In each of these cases, and similar instances involving other types of device files, a ZC audit record is written if auditing is on and the device file (*FILE object) is being audited. However, since no actual modification to the device file object is being done by the operating system during the I/O operation, the object change date is not updated on the *FILE object even though a ZC audit record is written.

The operating system will update the object change date for many reasons such as:

- When a user profile has private authority to an object, and that object is then deleted, the system updates the Change Date/Time field of that user profile as it removes that private authority.
- If security auditing is on when the object is deleted, a DO (Delete Operation) audit record is written for the deleted object.
- Because the system automatically updates every user profile that has private authority to the deleted object, no audit records are written for those user profiles, even though their Change Date/Time fields are updated.
- When internal updates are made to the object at runtime. These could include runtime statistics, object conversion, extending the size of an object during use in order to hold additional information, etc. These types of object updates will normally not result in a security audit record being sent to the QAUDJRN audit journal.

To track when your users have used normal system interfaces to change objects, use the security auditing journal. Reports to detect changes to objects that are based solely on the Change Date/Time field of an object can only produce partial results.

Why you should not use the Date/Time field for general security auditing

The main guideline used to decide what to audit for IBM i is to audit the security-relevant actions of users. The second guideline is to not write audit records for operations that the operating system automatically performs. In some cases, those automatic operations might be audited if the operating system performs the operation by using a function that is also designed to be used by users.

The objectives for maintaining the Change Date/Time field of an object are different from the audit objectives. The main purpose of the Change Date/Time field is to indicate when an object is changed. An updated Change Date/Time field does not indicate what was changed for the object or who made the change. One of the main uses of this field is to indicate that the object should be saved by the Save Changed Objects (SAVCHGOBJ) command. The SAVCHGOBJ command does not need to know when the last change was made, only that the object was changed since it was last saved. This feature allows performance to be optimized for database files. The Change Date/Time field is updated only the first time the file is changed after it was last saved. Performance can be affected if the Change Date/Time field was updated each time a record in the file was updated, added, or deleted.

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 893. This section 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

Not all of the authority failure and integrity violation messages are found in the QHST log. These messages are listed here.

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 IBM i 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.

- CPF2220. These events can be captured in the audit journal by specifying *AUTFAIL for the QAUDLVL system value.
- CPF4AAE. These events can be captured in the audit journal by specifying *AUTFAIL for the QAUDLVL system value.
- CPF2246. 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 might want more detailed information about the specific data that was accessed and changed. Object journaling can provide you with those details. Object journaling is used primarily for object integrity and recovery. Refer to the Journal management topic 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, user, and the time of access
- Before- and after-images of all object changes
- Records of when the object was opened, closed, changed, saved, created, deleted, and so on.

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 a database file, data area, data queue, library, or integrated file system object, you can use the **DSPJRN** command to print all the changes for that particular object. Here are some examples:

```
Type the following command for a particular database file.
DSPJRN JRN(library/journal) +
       FILE(library/file) OUTPUT(*PRINT)

Type the following command for a particular data area.
DSPJRN JRN(library/journal) +
       OBJ((library/object name *DTAARA))  OUTPUT(*PRINT)

Type the following command for a particular data queue.
DSPJRN JRN(library/journal) +
       OBJ((library/object name *DTAQ)  OUTPUT(*PRINT)

Type the following command for a particular integrated file system object.
DSPJRN JRN(library/journal) +
       OBJPATH(('path name')) OUTPUT(*PRINT)

Type the following command for a particular library.
DSPJRN JRN(library/journal) +
       OBJ(*LIBL/library-name *LIB) OUTPUT(*PRINT)
```

For example, if journal JRNCUST in library CUSTLIB is used to record information about file CUSTFILE (also in library CUSTLIB), the command can be:

```
DSPJRN JRN(CUSTLIB/JRNCUST) +
       FILE(CUSTLIB/CUSTFILE) OUTPUT(*PRINT)
```

You can also create an output file and do a query or use SQL to select all of the records from the output file for a specific output.

Type the following command to create an output file for a particular database file.

```
DSPJRN JRN(library/journal) +
                  FILE(library/file name) +
               OUTPUT(*OUTFILE) OUTFILEFMT(*TYPE5) OUTFILE(library/outfile) ENTDTALEN(*CALC)
```

Type the following command to create an output file for a particular data area.

```
DSPJRN JRN(library/journal) +
                  OBJ((library/object name *DTAARA)) +
               OUTPUT(*OUTFILE) OUTFILEFMT(*TYPE5) OUTFILE(library/outfile) ENTDTALEN(*CALC)
```

Type the following command to create an output file for a particular data queue.

```
DSPJRN JRN(library/journal) +
                  OBJ((library/object name *DTAQ)) +
               OUTPUT(*OUTFILE) OUTFILEFMT(*TYPE5) OUTFILE(library/outfile) ENTDTALEN(*CALC)
```

Type the following command to create an output file for a particular integrated file system object.

```
DSPJRN JRN(library/journal) +
                  OBJPATH(('path name')) +
               OUTPUT(*OUTFILE) OUTFILEFMT(*TYPE5) OUTFILE(library/outfile) ENTDTALEN(*CALC)
```

Type the following command to create an output file for a particular library.

```
DSPJRN JRN(library/journal) +
                  OBJ((*LIBL/library-name *LIB)) +
               OUTPUT(*OUTFILE) OUTFILEFMT(*TYPE5) OUTFILE(library/outfile) ENTDTALEN(*CALC)
```

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.

Related information

Journal management

Analyzing user profiles

You can display or print a complete list of all the users on your system by using the Display Authorized Users (**DSPAUTUSR**) command.

The list can be sequenced by profile name or group profile name. Here 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 UPSPAU field 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 output file) is equal to *NO or *PARTIAL.
- List all users who have a particular initial menu or initial program.
- List inactive users by looking at the date last sign-on field.
- 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 output file) 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 output file) is equal to Y.

Examining large user profiles

You might want to evaluate the security effectiveness of large user profiles on your system. User profiles with large numbers of authorities, appearing to be randomly spread over most of the system, can reflect a lack of security planning.

Here 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)
   ```

   Note: Directories and directory-based objects are not printed. WRKOBJOWN and WRKOBJPVT commands can be used to display directory-based objects and library-based objects, but there is no print function associated with these commands.

   Some IBM-supplied user profiles are very large because of the number of objects they own. Listing and analyzing them is 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” on page 312.

Related reference

IBM-supplied user profiles

This section 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.

Analyzing object and library authorities

You can audit the object and library authorities on your system.

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. You can analyze these programs to audit the security of the system.

The following method can be used to find and inspect those programs that adopt authority:

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 311 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)
```

   Note: The object type parameter might need to be *PGM, *SQLPKG, or *SRVPGM as indicated by the DSPPGMADP report.

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 adopted authority 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)
```

Note: The object type parameter might need to be *PGM, *SQLPKG, or *SRVPGM as indicated by the DSPPGMADP report.

Checking for objects that have been altered

An altered object is often an indication that someone is attempting to tamper with your system. You can use the Check Object Integrity (**CHKOBJITG**) command to check those objects that have been altered.

You might 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 have 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 check if libraries and commands have been tampered with. You can also start an integrated file system scan or check if objects failed a previous integrated file system scan.

Running the **CHKOBJITG** command requires *AUDIT special authority. The command might take a long time to run because of the scans and calculations that it performs. You should run it at a time when your system is not busy. Most IBM commands duplicated from a release before V5R2 will be logged as violations. These commands should be deleted and re-created using the Create Duplicate Object (**CRTDUPOBJ**) command each time a new release is loaded.

Related information

Scanning support

Checking the operating system

You can use the Check System (QYDOCHKS) API to check if any key operating system object has been changed since it was signed.

Any object that is not signed or has been changed since it was signed will be reported as an error. Only signatures from a system trusted source are valid.

Running the QYDOCHKS API requires *AUDIT special authority. The API might take a long time to run because of the calculations it performs. You should run it at a time when your system is not busy.

Related reference

Check System (QYDOCHKS) API

Auditing the security officer's actions

You can keep a record of all actions performed by users with *ALLOBJ and *SECADM special authority for tracking purpose.

To do this, you can use the action auditing value in the user profile:

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 or QAUDLVL2 system values 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)
```

   "Action auditing" on page 118 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.

   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 305.

Chapter 10. Authority collection

Authority collection is a capability that is provided as part of the base operating system. At a high level, authority collection captures data that is associated with the runtime authority checking that is built into the IBM i system. This data is logged to a repository provided by the system and interfaces are available to display and analyze the data. The intent of this support is to assist the security administrator and application provider in securing the objects in an application with the lowest level of authority that is required to allow the application to run successfully. By using the authority collection capability to remove or avoid excess authority, the overall security of the objects that are used by an application is improved.

Applications available for the IBM i server often have excessive authority that is granted to the objects within the application. Analysis of applications proves that this excessive authority setting is true today even with the current laws and regulations that require sensitive data to be adequately secured. Traditionally, the public authority (*PUBLIC) of objects within an application is set to an authority value that exceeds the authority that is required to run the application. For example, the public authority on a Db2 table object (*FILE) can be set to *CHANGE authority even though the application requires *USE authority to the data. This excessive authority setting opens a security exposure in the system as the data in this particular table object can be changed, outside of the application, by users of the system. Further analysis of the application security settings shows where the authority setting is even greater than *CHANGE authority. For some applications, the authority setting of *ALL is used which allows users of the system to change the object and data and even delete the entire object from the system. The authority collection support is designed to provide the security administrator and application provider a tool to help lock down the security of the application objects.

Interfaces are provided to allow a security administrator to collect and analyze data that is associated with the authority checking support of IBM i. These interfaces support the ability to start authority collection for a specific user of the system or for specific objects on the system.

The data that is collected during the application's runtime authority checks is significant in both volume and detail. For this reason, you must consider the performance impact that authority collection has on the runtime performance of an application. While the authority collection can be run on a production partition, the recommendation initially is to run the authority collection on a test partition where the application's runtime performance requirements are not the same as the production environment. In addition, changes made to the authority settings of the objects, based on the authority collection data, need to be fully tested before the authority changes are made in the production environment.

Authority checking support is built into the IBM i Operating System (OS) and Licensed Internal Code (LIC). Each authority check that is requested by the OS and LIC is logged to an authority collection data repository. Access to any IBM i object (*FILE, *PGM, *CMD, and other object types) requires the authority check to succeed before access to the object and data is allowed. For the authority check to succeed, the user, the user's groups, public authority, and program adopted authority settings are considered when the system checks for authority. Each object type can have different internal implementations and thus have different authority checking requirements. This is an important detail in relation to authority collection. For a single IBM i OS interface (CL Command, API, Service) numerous authority checks can occur against the object. Consider a simple example of calling a CL program that runs a simple command such as DSPJOBD or CHGJOBD. The system needs to find the library that contains the object, find the object within the library, lock the job description to prevent deletion while the interface is running, access the object itself to read (or change) the object and then display or change the data associated with the interface. Each of these steps, including locking the object, might perform an authority check against the object to make sure that the user is authorized to use the interface and target object. In fact, it is common that multiple authority checks are made by the OS and LIC for an object within a single CL command or API interface. The reason for this is that the authority checking logic that is built into the OS and LIC is run for internal interfaces that are used by the OS to access the object as well as the authority checks built into the interface itself.

An entry is logged in the authority collection repository for each unique authority check against the objects involved. This is important to understand as the authority that is required to the object must be

derived from the cumulative “required authority” value from all of the authority collection entries that are logged for the object. For more information, see Analyze the authority collection data.

When authority collection for a user is active, authority information is collected for objects that are accessed by this user. When this user runs a job on the system (interactive, batch, communication, and other types) and accesses objects within the application, authority collection data is gathered and written to the authority collection repository for the user.

When authority collection for objects is active and specific objects have an authority collection value other than *NONE, authority information is collected for these objects when accessed by any user. When a job is run on the system (interactive, batch, communication, and other types) which accesses these objects within the application, authority collection data is gathered for these objects and is written to the authority collection repository for objects.

Authority collection interfaces

There are several interfaces available for the authority collection support.

Authority collection for a user

- Start Authority Collection (STRAUTCOL) command.
- End Authority Collection (ENDAUTCOL) command.
- Delete Authority Collection (DLTAUTCOL) command.
- The authority collection active indicator and the authority collection repository exists indicator are shown by the following interfaces:
  - Display User Profile (DSPUSRPRF) command, *BASIC display, printed output, and outfile (QADSPUPB).
  - Dump User Profile (DMPUSRPRF) command (only authority collection active indicator).
  - Retrieve User Profile (RTVUSRPRF) command.
  - QSYS2.USER_INFO view.
- The Start Authority Collection (STRAUTCOL) command parameters from the most recent use of STRAUTCOL are shown by the following interfaces. These values are only shown if an authority collection repository currently exists for the user.
  - Display User Profile (DSPUSRPRF) command, *BASIC display and printed output.
  - Retrieve User Information (QSYRUSRI) API.
- IBM Navigator for i, Users and Groups function, contains support for authority collection for a user.
- QSYS2.AUTHORITY_COLLECTION view, display and analyze the authority collection data.

Authority collection for objects

- Start Authority Collection (STRAUTCOL) command.
- End Authority Collection (ENDAUTCOL) command.
- Delete Authority Collection (DLTAUTCOL) command.
- Change Authority Collection (CHGAUTCOL) command.
- The authority collection for objects active indicator is shown by the following interfaces:
  - Display Security Attributes (DSPSECA) command, display and printed output.
  - Retrieve Security Attributes (QSYRTVSA) API.
- The object's authority collection value is shown by the following interfaces:
  - Display Object Description (DSPOBJD) command, *FULL display, printed output, and outfile (QADSPOBJ).
  - Display Attributes (DSPATR) command, *FULL display and printed output.

  - Display Link (DSPLNK) command, when specifying option 8 to display attributes.
  - Retrieve Object Description (RTVOBJD) command.
  - List Objects (QUSLOBJ) API, format OBJL0700.
  - Open List of Objects (QGYOLOBJ) API, key 300 and 315.
  - Retrieve Object Description (QUSROBJD) API, format OBJD0400.
  - QSYS2.OBJECT_STATISTICS table function.
- The authority collection information is displayed and can be analyzed by the following interfaces:
  - QSYS2.AUTHORITY_COLLECTION_OBJECT view - for libraries and objects in libraries.
  - QSYS2.AUTHORITY_COLLECTION_LIBRARIES view - for all libraries and objects in all libraries.
  - QSYS2.AUTHORITY_COLLECTION_FSOBJ view - for file system objects in the "root" (/), QOpenSys, and user-defined file systems.
  - QSYS2.AUTHORITY_COLLECTION_DLO view - for document and folder objects.

  Note: QSYS2.AUTHORITY_COLLECTION_OBJECT and QSYS2.AUTHORITY_COLLECTION_LIBRARIES return the same results. However, QSYS2.AUTHORITY_COLLECTION_OBJECT will perform better when the number of entries in the authority collection is large and you are looking for a specific object or objects in a specific library. QSYS2.AUTHORITY_COLLECTION_LIBRARIES will perform better when the number of entries in the authority collection is small or you are looking for all or most objects in the authority collection.
- Objects in libraries with an authority collection value of *OBJINF are shown by the following table function:

```
SELECT * FROM TABLE (QSYS2.OBJECT_STATISTICS('*ALLUSR ','*ALL') ) AS X
  WHERE AUTHORITY_COLLECTION_VALUE = '*OBJINF'
```

- File system objects in the "root" (/), QOpenSys, and user-defined file systems with an authority collection value of *OBJINF are shown by doing the following:
  1. Run the Retrieve Directory Information (RTVDIRINF) command specifying the desired directory. This will produce a QAEZDxxxxO file. The RTVDIRINF command may take a long time to run.
  2. Use the QAEZDxxxxO file created by the RTVDIRINF command in the Select statement:

```
SELECT QEZOBJNAM, QEZOBJTYPE, QEZAUTCOL FROM QUSRSYS.QAEZDxxxxO
  WHERE QEZAUTCOL = '*OBJINF'
```

- IBM Navigator for i, Security function and File Systems function contain support for authority collection for objects.

Start authority collection

Authority collection can be started for a user or for objects. The same data is collected for authority collection for objects and for authority collection for a user (when authority collection is started for the user with DETAIL(*OBJINF)).

The difference between authority collection for a user and authority collection for objects

- Authority collection for a user collects authority information for authority checks on objects that are performed when a job is running under the specified user.
- Authority collection for objects collects authority information for all authority checks on the specified objects regardless of the user the job is running under.

Starting authority collection for a user

Authority collection by user means that the authority collection is only active for the “current user profile” of the job (the thread effective user profile). Authority collection can be active for multiple users at the same time and an authority collection repository exists for each user. By default, the data that is collected is object level authority data for the user. Object level authority data is defined as private authorities for a user to an object (including authorities from an authorization list), group profile authority information, public authority, and program adopted authority. The intent of this support is to allow the customer to better secure their data objects with object level authority settings.

Starting authority collection for a group user profile can be done but the authority collection for this user takes effect only when the user profile (the group profile in this case) is the “current user profile” of the job (essentially, from an authority checking standpoint, the user profile is not a group profile in this situation). For example, if USR1 has a group profile of GRP2, and authority collection is started for GRP2, no authority data is logged when user USR1 is the current user of the job and GRP2 is in the group profile list. Authority collection for user profile GRP2 occurs if GRP2 is the current user of the job. In addition, starting authority collection for a user profile that owns a program or service program that adopts owner authority does not have authority data logged (unless this user profile is the current user of the job). For example, user profile OWN1 owns a program that is called PGM1 and this program adopts owner authority (OWN1 is the program owner). If STRAUTCOL is run for user profile OWN1, and PGM1 is called by user USR1, no authority data is logged under the OWN1 authority collection repository. If USR1 is specified on STRAUTCOL, the authority collection data would be logged for program PGM1, including the information that PGM1 adopts the owner’s authority. For group profile and adopted authority situations, significant authority collection information is logged to the authority collection repository of the current user when either the group or adopting program owner is used to satisfy an authority check.

The Start Authority Collection (STRAUTCOL) command specifying TYPE(*USRPRF) is used to start the authority collection for a specified user profile. The command provides options to collect information for objects in libraries, document library objects (*DOC and *FLR object types), and objects in the "root" (/), QOpenSys, and user-defined file systems.

For objects in libraries, you can select which libraries, objects (including generic names), and object types to include in the authority collection for the specified user. In addition, an Omit Library (OMITLIB) parameter is available to omit certain libraries and corresponding objects from the authority collection.

For document library objects and file system objects, STRAUTCOL provides an option to include information only about specific object types. While the collection itself cannot be restricted to particular objects, folders, or directories, the interfaces provided for analyzing a collection are fully capable of selecting and reporting data only for specific objects of interest.

The Detail (DETAIL) parameter on the STRAUTCOL command specifies the details that are used to determine whether an authority check is for a unique instance. One unique instance is collected for each check. The *OBJINF value indicates that the authority checking information is collected for each unique instance of the object level information that is associated with the authority check. Specifying this value results in the collection of object level unique authority checks regardless of the job that accesses the object and regardless of the unique code paths within the job. The *OBJJOB value indicates that the authority checking information is collected for each unique instance of the object level information that is associated with the authority check and each unique instance of the job information that is associated with the authority check. Specifying this value results in the collection of object and job level unique authority checks plus each unique code path within the job is collected. For examples, see the Start Authority Collection (STRAUTCOL) command.

Authority collection for a specified user can be started by using the STRAUTCOL TYPE(*USRPRF) command and ended by using the ENDAUTCOL TYPE(*USRPRF) command. Authority collection can be restarted for a user after it is ended by using the STRAUTCOL TYPE(*USRPRF) command. This provides the capability to collect more authority data when the authority collection is restarted.

To collect authority information for the users that an application runs under:

1. Start authority collection for the user the application runs under. If the application runs under different users, then start authority collection for each user.

- `STRAUTCOL TYPE(*USRPRF) USRPRF(up_name) ...`

2. Run the application.
3. End authority collection for each user.
   - `ENDAUTCOL TYPE(*USRPRF) USRPRF(up_name)`
4. Analyze the authority data that is collected for each user.
5. Delete the authority collection data when it is no longer needed.
   - `DLTAUTCOL TYPE(*USRPRF) USRPRF(up_name)`

Starting authority collection for objects

Authority collection by object occurs when an object has an authority collection value other than *NONE and authority collection for objects is active. The data that is collected is object level authority data for the user running at the time the authority check is performed on the object. Object level authority data is defined as private authorities for a user to an object (including authorities from an authorization list), group profile authority information, public authority, and program adopted authority. The intent of this support is to allow the customer to better secure their data objects with object level authority settings.

For information about an object's authority collection value, see Change an object's authority collection value.

The Start Authority Collection (STRAUTCOL) command specifying TYPE(*OBJAUTCOL) is used to start authority collection for objects. Authority information is collected for objects with an authority collection value other than *NONE. An object's authority collection value is set by using the Change Authority Collection (CHGAUTCOL) command. Authority collection for objects is ended by using the End Authority Collection (ENDAUTCOL) command specifying TYPE(*OBJAUTCOL). Ending authority collection for objects does not change the object's authority collection value. Authority collection can be restarted for objects after it is ended by using the STRAUTCOL TYPE(*OBJAUTCOL) command.

To collect authority information for objects that an application uses:

1. Change the authority collection value for the desired objects to *OBJINF.
   - `CHGAUTCOL OBJ('/QSYS.LIB/MYLIB.LIB/MYOBJ.DTAARA') AUTCOLVAL(*OBJINF) ...`
   - `CHGAUTCOL OBJ('/path/obj') AUTCOLVAL(*OBJINF) ...`
2. Start authority collection for objects.
   - `STRAUTCOL TYPE(*OBJAUTCOL) ...`
3. Run the application.
4. End authority collection for objects.
   - `ENDAUTCOL TYPE(*OBJAUTCOL)`
5. Analyze the authority data that is collected for each object.
6. Change the authority collection value of the desired objects to *NONE to indicate that authority information is no longer collected.
   - `CHGAUTCOL OBJ('/QSYS.LIB/MYLIB.LIB/MYOBJ.DTAATA') AUTCOLVAL(*NONE) ...`
   - `CHGAUTCOL OBJ('/path/obj') AUTCOLVAL(*NONE) ...`
7. Delete the authority collection data for the objects when it is no longer needed.
   - `DLTAUTCOL TYPE(*OBJ) OBJ('/QSYS.LIB/MYLIB.LIB/MYOBJ.DTAATA') ...`
   - `DLTAUTCOL TYPE(*OBJ) OBJ('/path/obj') ...`

Change an object's authority collection value

When authority collection for objects is active, an object's authority collection value determines whether authority information is collected for the object.

The Change Authority Collection (CHGAUTCOL) command is used to change an object's authority collection value.

The authority collection value (AUTCOLVAL) parameter on the CHGAUTCOL command specifies whether to collect authority information for the object. A value of *NONE indicates that authority information is not collected for the object. A value of *OBJINF indicates that the authority checking information is collected for each unique instance of the object level information that is associated with the authority check. Specifying this value results in the collection of object level unique authority checks regardless of the job that accesses the object and regardless of the unique code paths within the job. For examples, see the Change Authority Collection (CHGAUTCOL) command.

If you are changing the authority collection value for a directory or a library, the CHGAUTCOL command provides a subtree (SUBTREE) parameter to indicate whether to also change the authority collection value for the objects in the directory or library.

If you are changing the authority collection value for a physical file, the CHGAUTCOL command provides a parameter to include dependent objects (INCDEPOBJ). This parameter indicates whether to also change the authority collection value for the logical files dependent on the data in the physical file.

If you are changing the authority collection value for a symbolic link, the CHGAUTCOL command provides a symbolic link (SYMLNK) parameter to indicate whether to change the symbolic link or the object pointed to by the symbolic link. If a symbolic link object is encountered, either specified in the Object (OBJ) parameter or encountered in the processing of a subtree, the value that is specified for the SYMLNK parameter is applied to that symbolic link object. If processing a subtree, the processing of that branch of the subtree then stops because a symbolic link object itself cannot have subtrees.

Authority collection repository damage

Damage can occur to the authority collection repository for a user or for objects.

The damage can frequently occur during an abnormal IPL of the partition where authority collection is active for users or for objects. For performance reasons, authority collection data is not immediately written out to disk when it is collected. Forcing the data to disk would result in unacceptable performance for the authority collection due to the volume and frequency of data that is written to the repository.

Unfortunately, damage to a user's or objects authority collection repository results in the loss of the previously collected authority data. A Db2 table object can be created at any time from the active authority collection data. This creates a "snapshot" of the data. If authority collection is run for an extended period, a table object can be periodically created and updated to prevent data loss if an abnormal IPL occurs.

Authority collection for a user

If an abnormal IPL occurs when authority collection for a user is active, the recovery is to delete the authority collection repository for the user. For each user, use the Delete Authority Collection (DLTAUTCOL) command specifying TYPE(*USRPRF) and then start the authority collection again.

To determine which user authority collection repositories need to be deleted, use the following SQL query:

```
SELECT AUTHORIZATION_NAME, AUTHORITY_COLLECTION_ACTIVE FROM
    QSYS2.USER_INFO WHERE
    AUTHORITY_COLLECTION_REPOSITORY_EXISTS='YES';
```

Before a user authority collection repository can be deleted by using the DLTAUTCOL command, authority collection for the user must first be ended by using the End Authority Collection (ENDAUTCOL) command.

Use the AUTHORIZATION_NAME values returned by the query on the ENDAUTCOL and DLTAUTCOL commands.

Authority collection for objects

During an IPL, the system checks whether the authority collection repository for objects is damaged. If so, the authority collection repository is automatically deleted and authority collection for objects is restarted if it was previously active. If the authority collection repository for objects is damaged while the partition is active, end authority collection for objects by using the ENDAUTCOL command. Use the DLTAUTCOL TYPE(*OBJ) OBJ(*ALL) command to delete the common authority collection repository for all objects, and then start the authority collection again.

Save and restore considerations

The Authority collection data repository for a user or objects is not saved or restored.

The authority collection active indicator in the user profile is saved and restored.

The indicator of whether authority collection for objects is active is not saved or restored.

The authority collection value in the object is not saved or restored. When an object is restored that currently exists on the system, the authority collection value for the object on the system remains unchanged.

Authority collection repository

The Save Security Data (SAVSECDTA) command and any other save interface, does not have support to save the authority collection data for a user or objects.

To save the authority collection data, it must first be written to a Db2 table (*FILE object) by querying the view. See Display authority collection data for an example of writing the authority collection data to a table. The Db2 table object can then be saved and restored if necessary.

Authority collection active indicator in the user profile

The authority collection active indicator in the user profile is saved for each profile when the SAVSECDTA command is used.

When the Restore User Profile (RSTUSRPRF) command is used to restore a user profile, the authority collection active indicator is restored as follows:

- If the profile on the media has authority collection active then a check is made to see whether the authority collection repository for the user exists on the system. If it does, then the restored user profile has authority collection active. If it does not, then the restored user profile has authority collection turned off with the End Authority Collection (ENDAUTCOL) command.
- If the profile on the media does not have authority collection active, then the restored user profile does not have authority collection active.

Special considerations for authority collection

1. The authority collection support does NOT collect data that is related to interfaces that check special authority. Authority collection data that is related to *ALLOBJ special authority is collected as it affects object level security. Other special authority checks, such as *JOBCTL or *SAVSYS, do not generate authority collection entries. Special authority settings for a specific user profile are easy to check by using the existing security interfaces such as the Display User Profile (DSPUSRPRF) command and related APIs or by querying the QSYS2.USER_INFO view.
2. Function usage settings (also called application administration) are not collected for the same reason as special authority settings. Function usage settings for a specific user profile are easy to check and

are managed by using the Work with Function Usage (WRKFCNUSG) command or by querying the QSYS2.FUNCTION_USAGE view.

3. The system automatically excludes authority collection data when the IBM i operating system accesses an object and authority is available because of program adopted authority from the operating system. The operating system uses program adopted authority to manage and secure objects and control blocks that it uses. In addition, the operating system uses program adopted authority for situations where it requires access to an object for a specific reason and the current user of the job is not authorized.
4. The open file (*FILE objects) support for authority collection is for full opens only (no shared or pseudo open is logged). The initial authority collection occurs at file open but the data is not written to the authority collection repository until a hard close on the file is done. Writing the authority collection data to the repository for the file open/close case must be done at close time to accurately log the authority that is required for the application. The open might be done for read/add/update/delete but the application might only read the data.
5. Authority collection of column permissions for a Db2 table is not supported.
6. If an authority collection contains information for an object that resides in an Independent Auxiliary Storage Pool (IASP) then that IASP must be available when a query is run against the collection. If the IASP is not available, the information for that object will not be included in the query results.

Special considerations for authority collection for a user

1. The system automatically excludes certain system libraries and their objects, such as QRCL, QRECOVERY, QSPL, QTEMP, QPTFOBJ1, or QPTFOBJ2 (and the corresponding IASP version of the system libraries), from the authority collection data. Also excluded are authority checks against objects that are not in a library, folder, or directory.
2. The system automatically excludes IBM i programs and service programs from the authority collection data. Programs or service programs that are *SYSTEM domain or have a program state of *SYSTEM or *INHERIT are excluded from the authority collection. These attributes can be displayed by using the Display Program (DSPPGM) and Display Service Program (DSPSRVPGM) commands.
3. If the STRAUTCOL command is used to start the authority collection for a user profile and the partition is IPLed, the authority collection continues when a job (post IPL) running under the specified user profile starts.
4. The system automatically excludes authority collection data for document library objects and file system objects that have been deleted.
5. IBM i supports a capability that is called profile swap. A profile swap can occur within an active job to swap the current user of a thread from one user to another. When this profile swap occurs, the authority collection of the previous user, for this thread, is no longer active because the current user changed. If the newly swapped user has authority collection active, any authority checks made are now logged under this user's authority collection repository.
6. If a user profile with an active authority collection is deleted, the authority collection is automatically ended before the user profile is deleted.
7. To collect authority information for object types that are only allowed in QSYS (for example, *LIB), specify parameter LIBINF(*ALL) on the STRAUTCOL command. When authority collection includes object type *LIB, library objects that start with QSYS* are automatically excluded from the authority collection data.
8. When authority collection is started for a user that has an existing authority collection data repository, new authority data is added to the existing information unless parameter DLTCOL(*YES) is specified. New authority collection data can only be added to the existing information if the value specified on the DETAIL parameter matches the value that was specified on the DETAIL parameter when the existing authority information was collected.

Special considerations for authority collection for objects

1. If the STRAUTCOL command is used to start authority collection for objects and the partition is IPLed, the authority collection remains active after the IPL.
2. The authority collection value cannot be set for QTEMP library or objects in it.
3. The authority collection value cannot be set for QSYS library or libraries with names that begin with QSYS.
4. When changing the authority collection value by specifying an object name pattern in the OBJ parameter or by specifying SUBTREE(*ALL) any objects of an unsupported type are ignored.
5. When authority collection for objects is started and data exists in the authority collection repository for objects, new authority data is added to the existing information unless parameter DLTCOL(*ALL) is specified.
6. When the operating system is installed, authority collection for objects is ended if it is active.

End authority collection

Authority collection can be ended for a specified user or for all objects on the partition.

The End Authority Collection (ENDAUTCOL) command specifying TYPE(*USRPRF) and the user profile name stops the authority collection for the specified user. The ENDAUTCOL command must be run after all jobs that are running under the specified user have ended to ensure that all of the information for this user is collected.

The ENDAUTCOL TYPE(*OBJAUTCOL) command stops the authority collection for objects. The authority collection value on the objects is not changed.

For Db2 objects of type *FILE, collecting authority information occurs during file open, subsequent file I/O, and the file close. A full close of the *FILE must be done for complete authority information to be collected for the object.

Authority collection can be started by using the STRAUTCOL command and ended by using the ENDAUTCOL command. Authority collection can be restarted after it has ended by using the STRAUTCOL command. This provides the capability to collect more authority data when the authority collection is restarted.

Ending authority collection does not delete the authority collection repository. The data remains in the repository until it is removed or the repository is deleted.

Delete authority collection repository

The authority collection repository for a user can be deleted. The authority collection repository for objects can be deleted or information for an object or group of objects can be deleted.

To save the authority collection data before DLTAUTCOL is used, it must first be written to a Db2 table (*FILE object) by using the provided view support. See Display authority collection data for an example of writing the authority collection data to a table.

Authority collection for a user

The Delete Authority Collection (DLTAUTCOL) command specifying TYPE(*USRPRF) and a profile name deletes the authority collection repository for the specified user. Deleting the authority collection repository deletes all authority collection information for the specified user. The authority collection repository can also be deleted when the Start Authority Collection (STRAUTCOL) command is run by using the DLTCOL(*YES) parameter.

Authority collection for objects

The DLTAUTCOL command specifying TYPE(*OBJ) and an object name deletes the authority collection information for the specified object or group of objects. The object repository is not deleted because

it is a common repository for all information when collecting authority information for objects. The common object repository can be deleted with the DLTAUTCOL TYPE(*OBJ) OBJ(*ALL) command. The common object repository can also be deleted when the Start Authority Collection (STRAUTCOL) TYPE(*OBJAUTCOL) command is run by using the DLTCOL(*ALL) parameter. Deleting the common object repository deletes the authority collection information for all objects.

Display authority collection data

Authority collection captures a significant amount of information that is associated with the authority checking of an object. SQL views are used to display and analyze this information.

Authority collection for a user

The SQL view QSYS2.AUTHORITY_COLLECTION is used to display and analyze the authority information that was collected for a user.

IBM Navigator for i shows the authority collection information for a specific user but not in a form that can be queried. IBM Navigator for i has interfaces for authority collection for a user within the Users and Groups function.

- There are tasks in the console navigation area under Manage Collections to start, end, display, and delete authority collection for a user.
- There are tasks available for a user within the User list to start, end, display, and delete authority collection.
- An Authority Collection tab on the Capabilities page of the User properties panel shows the current authority collection status for the user.
- There is a table view of the items included in the authority collection. This can be viewed in a web table, or in a client viewer if IBM i Access Client Solutions (ACS) is installed on the PC. The web table also supports Properties and Permissions actions for each object that appears in the list.

The Run SQL Scripts function in ACS can be used to query the authority collection views. See the following SQL query examples that can be run against the view. Additional examples are built into ACS. Select the Insert from Examples feature and type "authority_collection" in the search bar.

Example queries that use the AUTHORITY_COLLECTION view

View authority collection data for USER1.

```
SELECT * FROM QSYS2.AUTHORITY_COLLECTION
    WHERE USER_NAME = 'USER1'
```

View authority collection data for USER1 for object PAYROLL in library PAYLIB.

```
SELECT * FROM QSYS2.AUTHORITY_COLLECTION
   WHERE USER_NAME = 'USER1' AND
   SYSTEM_OBJECT_NAME = ‘PAYROLL' AND SYSTEM_OBJECT_SCHEMA = ‘PAYLIB’
```

View authority collection data for USER1, object PAYROLL in PAYLIB, and object type *FILE.

```
SELECT * FROM QSYS2.AUTHORITY_COLLECTION
    WHERE USER_NAME = 'USER1' AND
    SYSTEM_OBJECT_NAME = 'PAYROLL'  AND SYSTEM_OBJECT_SCHEMA = ‘PAYLIB' AND
    SYSTEM_OBJECT_TYPE = '*FILE'
```

Example of saving the authority collection information for a user

Save the authority collection data for USER1 to Db2 table MYLIB.MYFILE. Writing the authority collection data to a Db2 table allows the data to be saved and restored to another partition. The Db2 table can then be analyzed by querying the resulting Db2 table.

```
CREATE TABLE MYLIB.MYFILE AS
    (SELECT * FROM AUTHORITY_COLLECTION WHERE USER_NAME = 'USER1') WITH DATA

SELECT * FROM MYLIB.MYFILE
```

Authority collection for objects

The following SQL views are used to display and analyze the authority information that was collected for objects:

- QSYS2.AUTHORITY_COLLECTION_OBJECT
- QSYS2.AUTHORITY_COLLECTION_LIBRARIES
- QSYS2.AUTHORITY_COLLECTION_FSOBJ
- QSYS2.AUTHORITY_COLLECTION_DLO

IBM Navigator for i shows the authority collection information for specific objects but not in a form that can be queried. IBM Navigator for i has interfaces for authority collection for objects.

- Within the File Systems function and the Security function there are tasks in the console navigation area under Authority Collection for Objects to manage authority collection for objects.
- Within an object list there are Authority Collection tasks for an object to change the authority collection value, display the information collected, and delete the information collected.
- The Security tab on the object's properties panel shows whether the object is currently included in the authority collection.

The Run SQL Scripts function in ACS can be used to query the authority collection views. See the following SQL query examples that can be run against the view.

Example queries that use the AUTHORITY_COLLECTION_OBJECT view

View data in the authority collection repository for objects, specific object (PAYROLL) of object type *FILE in library PAYLIB.

```
SELECT * FROM QSYS2.AUTHORITY_COLLECTION_OBJECT
    WHERE SYSTEM_OBJECT_SCHEMA = 'PAYLIB' AND
    SYSTEM_OBJECT_TYPE = '*FILE' AND SYSTEM_OBJECT_NAME = 'PAYROLL'
```

View data in the authority collection repository for objects, all objects of object type *FILE in library PAYLIB that begins with 'PAY'.

```
SELECT * FROM QSYS2.AUTHORITY_COLLECTION_OBJECT
    WHERE SYSTEM_OBJECT_SCHEMA = 'PAYLIB' AND SYSTEM_OBJECT_TYPE = '*FILE' AND
    SYSTEM_OBJECT_NAME like 'PAY%'
```

Example queries that use the AUTHORITY_COLLECTION_LIBRARIES view

View data in the authority collection repository for objects, all QSYS.LIB objects.

```
SELECT * FROM QSYS2.AUTHORITY_COLLECTION_LIBRARIES
```

View data in the authority collection repository for objects, all objects in selected libraries.

```
SELECT * FROM QSYS2.AUTHORITY_COLLECTION_LIBRARIES
   WHERE SYSTEM_OBJECT_SCHEMA IN ('MYLIB1', 'MYLIB2')
```

Example queries that use the AUTHORITY_COLLECTION_FSOBJ view

View data in the authority collection repository for objects, all objects in the "root" (/), QOpenSys, and user-defined file systems.

```
SELECT * FROM QSYS2.AUTHORITY_COLLECTION_FSOBJ
```

View data in the authority collection repository for objects, specific object in the "root" (/), QOpenSys, and user-defined file systems.

```
SELECT * FROM QSYS2.AUTHORITY_COLLECTION_FSOBJ WHERE PATH_NAME = '/mydir/mystmf'
```

Example queries that use the AUTHORITY_COLLECTION_DLO view

View data in the authority collection repository for objects, all document library objects (*DOC and *FLR object types).

```
SELECT * FROM QSYS2.AUTHORITY_COLLECTION_DLO
```

View data in the authority collection repository for objects, specific document object.

```
SELECT * FROM QSYS2.AUTHORITY_COLLECTION_DLO WHERE PATH_NAME = '/QDLS/QDIADOCS/NEWDOC'
```

Example of saving authority collection information for objects

Saving the authority collection data for objects requires three tables. Save the authority collection data to tables MYLIB.LIBOBJFILE, MYLIB.FSOBJFILE, MYLIB.DLOOBJFILE. Writing the authority collection data to Db2 tables allows the data to be saved and restored to another partition. The Db2 tables can then be analyzed by querying the resulting Db2 tables.

```
CREATE TABLE MYLIB.LIBOBJFILE AS (SELECT * FROM AUTHORITY_COLLECTION_LIBRARIES) WITH DATA
CREATE TABLE MYLIB.FSOBJFILE AS (SELECT * FROM AUTHORITY_COLLECTION_FSOBJ) WITH DATA
CREATE TABLE MYLIB.DLOOBJFILE AS (SELECT * FROM AUTHORITY_COLLECTION_DLO) WITH DATA
```

Analyze authority collection data

The authority collection data can be analyzed to help you secure the objects in an application.

The **detailed required authority** value that is returned in the **DETAILED_REQUIRED_AUTHORITY** field by the authority collection views is a key piece of information available to help the security administrator or application owner better secure the object. The detailed required authority value represents the authority that the system requires to pass the authority check against the object. By analyzing the detailed required authority value from every authority collection entry for a specific object, you can determine the minimum level of authority that can be granted to an object and still allow the application to run successfully.

To generate the authority collection entries, you must run the application to completion taking into account all code paths within the application. For example, if the application has special processing for end of quarter or year end, you must consider these code paths along with the normal runtime processing within the application. After the authority collection entries are generated, the detailed required authority values from the authority collection determine what authority the user needs to run the application successfully. If the detailed required authority value from all authority collection entries is less than the users current authority, the excess authority can be revoked for this user (or group or *PUBLIC) to set the authority to the lowest possible value and better secure the object.

Two authority collection values that are returned by the authority collection views, **DETAILED_CURRENT_AUTHORITY** and **DETAILED_CURRENT_ADOPTED_AUTHORITY**, provide the authority values available in the job at the time of the authority check. The authority available in the job comes from the user's authority, the authority from any group user profiles, public authority,

and adopted authority from the owner of currently running programs or service programs in the job. The **AUTHORITY_SOURCE** and **ADOPTED_AUTHORITY_SOURCE** values that are returned by the view indicate the source of the authority data that is logged in each authority collection entry.

Authority collection views

The information collected for an authority check on an object by authority collection for a user and by authority collection for objects can be looked at with views. The same information is collected for both types of collections but different views must be used to look at the information.

To access information using these views, the caller must have *ALLOBJ and *SECADM special authorities or be authorized to the QIBM_DB_SECADM function usage ID.

Authority collection for a user view

- AUTHORITY_COLLECTION - This view is used to look at information that was collected during authority collection for a user.

Authority collection for objects views

- AUTHORITY_COLLECTION_OBJECT - This view is used to look at information that was collected for libraries and objects in libraries during authority collection for objects.
- AUTHORITY_COLLECTION_LIBRARIES - This view is used to look at information that was collected for all libraries and objects in libraries during authority collection for objects.

  Note: QSYS2.AUTHORITY_COLLECTION_OBJECT and QSYS2.AUTHORITY_COLLECTION_LIBRARIES return the same results. However, QSYS2.AUTHORITY_COLLECTION_OBJECT will perform better when the number of entries in the authority collection is large and you are looking for a specific object or objects in a specific library. QSYS2.AUTHORITY_COLLECTION_LIBRARIES will perform better when the number of entries in the authority collection is small or you are looking for all or most objects in the authority collection.
- AUTHORITY_COLLECTION_FSOBJ - This view is used to look at information that was collected for all file system objects in the "root" (/), QOpenSys, and user-defined file systems during authority collection for objects.
- AUTHORITY_COLLECTION_DLO - This view is used to look at information that was collected for document library objects (DLO) during authority collection for objects.

Layout of authority collection views

The following table describes the columns in the views. The schema is QSYS2.

Table 135. All authority collection views

Column Name	System Column Name	Data Type	Description
AUTHORIZATION_NAME	USER_NAME	VARCHAR(10) Nullable	For the AUTHORITY_COLLECTION view, this is the name of the user profile for which authority information was collected. For the AUTHORITY_COLLECTION_OBJECT, AUTHORITY_COLLECTION_LIBRARIES, AUTHORITY_COLLECTION_FSOBJ, and AUTHORITY_COLLECTION_DLO views, this is the current user associated with the thread of the job in which the authority check was made.
CHECK_TIMESTAMP	CHKTIME	TIMESTAMP Nullable	The date and time the authority check was made.
SYSTEM_OBJECT_NAME	SYS_ONAME	VARCHAR(10) Nullable	The name of the object whose authority was checked. This field contains information for objects in libraries and document library objects (*DOC and *FLR object types). Document library objects in this field will be in *SYSOBJNAM format. File system objects and document library objects use the **PATH_NAME** field.
SYSTEM_OBJECT_SCHEMA	SYS_DNAME	VARCHAR(10) Nullable	The name of the library that contains the object.
SYSTEM_OBJECT_TYPE	SYS_OTYPE	VARCHAR(8) Nullable	The object type of the object.

Table 135. All authority collection views (continued)

Column Name	System Column Name	Data Type	Description
ASP_NAME	ASP_NAME	VARCHAR(10) Nullable	The name of the auxiliary storage pool to which storage for the object is allocated
ASP_NUMBER	ASP_NUMBER	DECIMAL(5,0) Nullable	The number of the auxiliary storage pool to which storage for the object is allocated. A value of 0 indicates *SYSBAS.
OBJECT_NAME	ONAME	VARCHAR(128) Nullable	The SQL name of the object. Objects supported by SQL may have the same name as the IBM i name or may have a different longer name than the IBM i name (SYSTEM_OBJECT_NAME).
OBJECT_SCHEMA	OSCHEMA	VARCHAR(128) Nullable	The SQL name of the schema (library). Schemas in SQL may have the same name as the IBM i name or may have a different longer name than the IBM i name (SYSTEM_OBJECT_SCHEMA).
OBJECT_TYPE	OTYPE	VARCHAR(9) Nullable	The SQL object type. The following values can be returned. • ALIAS - The object is an SQL alias. • FUNCTION - The object is an SQL function. • INDEX - The object is an SQL index. • PACKAGE - The object is an SQL package. • PROCEDURE - The object is an SQL procedure. • ROUTINE - The object is used in SQL by one or more external functions and/or external procedures. • SEQUENCE - The object is an SQL sequence. • TABLE - The object is an SQL table. • TRIGGER - The object is an SQL trigger. • TYPE - The object is an SQL type. • VARIABLE - The object is an SQL global variable. • VIEW - The object is an SQL view. • XSR - The object is an XML schema repository object.
AUTHORIZATION_LIST	AUTL	VARCHAR(10) Nullable	The name of the authorization list used to secure the object. This field contains data only if the object is secured by an authorization list
AUTHORITY_CHECK_SUCCESSFUL	CHKSUCCESS	CHAR(1) Nullable	The result of the authority check. This field is set to '1' if the authority check was successful and '0' if the authority check was not successful.
CHECK_ANY_AUTHORITY	CHKANYAUTH	CHAR(1) Nullable	Indicates whether the authority check that is performed by the system is for "ANY" of the authorities that are listed in the **DETAILED_REQUIRED_AUTHORITY** field. This field is set to '1' if "ANY" of the authorities were checked and '0' if specific authorities were checked. Certain authority checks allow the function to complete if the user associated with the currently running job has one or more of the authorities that are listed in the **DETAILED_REQUIRED_AUTHORITY** field. A common function that performs the "ANY" authority check is the system lock instruction that is used by many system commands, APIs, and services.
CACHED_AUTHORITY	CACHEAUTH	CHAR(1) Nullable	The operating system (OS) and Licensed Internal Code (LIC) have the capability to cache the authority the user currently has to an object, and use this authority for future authority checks. This field is set to '1' if authority was cached and '0' if authority was not cached. For performance reasons, the authority collection code will log, to the authority collection repository, the first authority check where cached authority is initially stored. Future authority checks, that use the cached authority, are not logged to the authority collection repository. However, any future authority check that requires more authority than was initially cached results in the logging of an authority collection entry for the authority check. In addition, the authority collection entries that have this field set to '1' might not always provide an accurate view of the required authority information. The reason for this is that the system code can cache the maximum authority the current user of the job has to the object but require only a subset of this authority to pass a future authority check. This is a rare case within the OS and LIC but might occasionally be done.
REQUIRED_AUTHORITY	REQAUTH	VARCHAR(7) Nullable	The authority that is required by the system to access the object. If the **DETAILED_REQUIRED_AUTHORITY** field does not map to a system-defined object authority level, this field will be blank. See "Authority field values" on page 331.
DETAILED_REQUIRED_AUTHORITY	DTLREQAUTH	VARCHAR(90) Nullable	The detailed individual authority values that are required by the system to access the object. This is an important piece of information in the authority collection data. The detailed required authority is what is used to determine what authority can be set on the object so that it passes the authority check. Analyzing all of the authority collection entries for an object indicate what authority value can be set on the object to allow the application to run successfully from an authority standpoint. See "Detailed authority field values" on page 332.
CURRENT_AUTHORITY	CURAUTH	VARCHAR(8) Nullable	The authority that the user currently has to the object. The **AUTHORITY_SOURCE** field must also be evaluated to determine where the users' authority to the object was found. If the **DETAILED_CURRENT_AUTHORITY** field does not map to a system-defined object authority level, this field will be blank. See "Authority field values" on page 331.
DETAILED_CURRENT_AUTHORITY	DTLCURAUTH	VARCHAR(99) Nullable	The detailed authority values that the user currently has to the object. The **AUTHORITY_SOURCE** field must also be evaluated to determine where the users' authority to the object was found. See "Detailed authority field values" on page 332.

Table 135. All authority collection views (continued)

Column Name	System Column Name	Data Type	Description
AUTHORITY_SOURCE	AUTHSRC	VARCHAR(50) Nullable	Where the system found the authority that either satisfied the authority check or caused the authority check to end unsuccessfully. • USER *ALLOBJ - All object special authority from the user • USER OWNERSHIP - User ownership • USER PRIVATE - User private authority • AUTHORIZATION LIST OWNERSHIP - Authorization list ownership • AUTHORIZATION LIST PRIVATE - Authorization list private authority • GROUP *ALLOBJ - Group profile all object special authority • GROUP OWNERSHIP - Group ownership • GROUP PRIVATE - Group private authority • PRIMARY GROUP - Primary group authority • AUTHORIZATION LIST GROUP OWNERSHIP - Authorization list group ownership • AUTHORIZATION LIST PRIMARY GROUP - Authorization list primary group authority • AUTHORIZATION LIST GROUP PRIVATE - Authorization list group private authority • AUTHORIZATION LIST PUBLIC - Authorization list public authority • PUBLIC - Public authority • Also see the **ADOPTED_AUTHORITY_SOURCE** field.
GROUP_NAME	GROUP_NAME	VARCHAR(10) Nullable	The name of the group profile whose authority was used to satisfy the authority check. If multiple group profiles contribute to the accumulated current authority for the object, this field contains the last group to contribute and the **MULTIPLE_GROUPS_USED** field is set to '1'. Group profiles are checked for authority based on the order in the group profile and supplemental group profile list in the user profile.
MULTIPLE_GROUPS_USED	MLTGRPUSED	CHAR(1) Nullable	Indicates whether multiple group profiles contributed to the **DETAILED_CURRENT_AUTHORITY** for the object. This field is set to '1' if multiple group profiles contributed and '0' if no group profiles or only one group profile's authority is used.
ADOPT_AUTHORITY_USED	ADOPTUSED	CHAR(1) Nullable	Indicates whether adopted authority is used to satisfy the authority check. This field is set to '1' if the authority of the adopting program owner is used to satisfy the authority check. This field is set to '0' if adopted authority was not used to satisfy the authority check. In addition, when this field is set to '0', the **ADOPTING_PROGRAM_NAME** field can contain the name of a program that is on the program invocation stack of the thread. If a program is listed, this program adopts the owners' authority and would satisfy the authority check if authority was not available from another authority source in the thread. That is, excessive authority could be removed, and adopted authority used. If no program name is listed in the **ADOPTING_PROGRAM_NAME** field, then this indicates no program in the invocation stack would satisfy the authority check for the object.
MULTIPLE_ADOPTING_ PROGRAMS_USED	MLTADOPTPG	CHAR(1) Nullable	Indicates whether the owners of multiple programs that adopt contribute authority to the combined **DETAILED_CURRENT_ADOPTED_AUTHORITY** field. This field is set to '1' if multiple programs that adopt contributed and '0' if no programs that adopt or only one program that adopts is used.
ADOPTING_PROGRAM_NAME	ADOPTPGM	VARCHAR(10) Nullable	The name of the program that adopts the owners' authority. If multiple adopting programs contribute to the accumulated **DETAILED_CURRENT_ADOPTED_AUTHORITY** for the object, the last program to contribute is listed and the **MULTIPLE_ADOPTING_PROGRAMS_USED** field is set to '1'. Adopting programs are checked for authority in order from the most recent invocation to the oldest invocation on the program invocation stack.
ADOPTING_PROGRAM_SCHEMA	ADOPTLIB	VARCHAR(10) Nullable	The name of the library that contains the adopting program.
ADOPTING_PROCEDURE_NAME	ADOPTPRC	VARCHAR(256) Nullable	The name of the adopting Integrated Language Environmet (ILE) program procedure.
ADOPTING_PROGRAM_TYPE	ADOPTPGMT	VARCHAR(8) Nullable	The object type of the adopting program.
ADOPTING_PROGRAM_ ASP_NAME	ADOPTPGMA	VARCHAR(10) Nullable	The name of the auxiliary storage pool to which storage for the adopting program is allocated.
ADOPTING_PROGRAM_ ASP_NUMBER	ADOPTPGMAN	DECIMAL(5,0) Nullable	The number of the auxiliary storage pool to which storage for the adopting program is allocated. A value of 0 indicates *SYSBAS.
ADOPTING_PROGRAM_ STATEMENT_NUMBER	ADOPTPGMSN	DECIMAL(10,0) Nullable	The statement number of the adopting program.
ADOPTING_PROGRAM_OWNER	ADOPTPGMOW	VARCHAR(10) Nullable	The name of the adopting program owner. The adopting program owners' authority is included in the authority checking algorithm of the system when the program in the **ADOPTING_PROGRAM_NAME** field is on the program invocation stack. **Note:** The ability to block adopted authority from previous invocations exists, by using the Use Adopted Authority attribute of a program. This attribute can be changed by using the Change Program (CHGPGM) command. When the Use Adopted Authority value of *NO is set on a program, this prevents any adopted authority from previous invocations from being included in the authority checking algorithm of the system.

Table 135. All authority collection views (continued)

Column Name	System Column Name	Data Type	Description
CURRENT_ADOPTED_AUTHORITY	CURADPT	VARCHAR(8) Nullable	The authority value that the adopting program owner currently has to the object. The **ADOPTED_AUTHORITY_SOURCE** field must also be evaluated to determine where the adopting program owners' authority to the object was found. If the **DETAILED_CURRENT_ADOPTED_AUTHORITY** field does not map to a system-defined object authority level, this field will be blank. See "Authority field values" on page 331.
DETAILED_CURRENT_ADOPTED_ AUTHORITY	DTLCURADPT	VARCHAR(99) Nullable	The detailed authority values that the adopting program owner currently has to the object. The **ADOPTED_AUTHORITY_SOURCE** field must also be evaluated to determine where the adopting program owners' authority to the object was found. See "Detailed authority field values" on page 332.
ADOPTED_AUTHORITY_SOURCE	ADOPTAUTSR	VARCHAR(50) Nullable	Where the system found the adopted authority that either satisfied the authority check or caused the authority check to end unsuccessfully. • ADOPTED *ALLOBJ - All object special authority from the adopting program owner. • ADOPTED OWNERSHIP - Adopted ownership from the adopting program owner. • ADOPTED PRIMARY GROUP - Adopted primary group authority from the adopting program owner. • ADOPTED PRIVATE - Adopted private authority from the adopting program owner. • ADOPTED AUTHORIZATION LIST OWNERSHIP - Adopted authorization list ownership from the adopting program owner. • ADOPTED AUTHORIZATION LIST PRIMARY GROUP - Adopted authorization list primary group authority from the adopting program owner. • ADOPTED AUTHORIZATION LIST PRIVATE - Adopted authorization list private authority from the adopting program owner.
MOST_RECENT_ PROGRAM_INVOKED	PGMINV	VARCHAR(10) Nullable	The name of the most recent program on the program invocation stack when the authority check was made.
MOST_RECENT_ PROGRAM_SCHEMA	PGMLIBINV	VARCHAR(10) Nullable	The name of the library that contains the most recent program invoked.
MOST_RECENT_ MODULE	MODINV	VARCHAR(30) Nullable	The name of the bound module within the most recently invoked ILE program.
MOST_RECENT_ PROGRAM_PROCEDURE	PGMPRC	VARCHAR(256) Nullable	The name of the most recently invoked ILE program procedure.
MOST_RECENT_ PROGRAM_TYPE	PGMTYP	VARCHAR(8) Nullable	The object type of the most recent program invoked.
MOST_RECENT_ PROGRAM_ASP_NAME	PGMASP	VARCHAR(10) Nullable	The name of the auxiliary storage pool to which storage for the most recent program is allocated.
MOST_RECENT_ PROGRAM_ASP_NUMBER	PGMASPN	DECIMAL(5,0) Nullable	The number of the auxiliary storage pool to which storage for the most recent program is allocated. A value of 0 indicates *SYSBAS.
MOST_RECENT_ PROGRAM_STATEMENT_NUMBER	PGMSTMN	DECIMAL(10,0) Nullable	The statement number of the most recent program.
MOST_RECENT_USER_STATE_ PROGRAM_INVOKED	USTPGM	VARCHAR(10) Nullable	The name of the most recent user state program on the program invocation stack when the authority check was made. A user state program is a program that is not part of the System State portion of the IBM i OS or the System State portion of an IBM product. Programs created by customers, programs created by application providers, and many products provided by IBM run in user state.
MOST_RECENT_USER_STATE_ PROGRAM_SCHEMA	USTLIB	VARCHAR(10) Nullable	The name of the library that contains the most recent user state program invoked.
MOST_RECENT_USER_STATE_ MODULE	USTMOD	VARCHAR(30) Nullable	The name of the bound module within the most recently invoked user state ILE program.
MOST_RECENT_USER_STATE_ PROGRAM_PROCEDURE	USTPGMPRC	VARCHAR(256) Nullable	The name of the most recently invoked user state ILE program procedure.
MOST_RECENT_USER_STATE_ PROGRAM_TYPE	USTPGMTYP	VARCHAR(8) Nullable	The object type of the most recent user state program invoked.
MOST_RECENT_USER_STATE_ PROGRAM_ASP_NAME	USTPGMASP	VARCHAR(10) Nullable	The name of the auxiliary storage pool to which storage for the most recent user state program is allocated.
MOST_RECENT_USER_STATE_ PROGRAM_ASP_NUMBER	USTPGMASPN	DECIMAL(5,0) Nullable	The number of the auxiliary storage pool to which storage for the most recent user state program is allocated. A value of 0 indicates *SYSBAS.

Table 135. All authority collection views (continued)

Column Name	System Column Name	Data Type	Description
MOST_RECENT_USER_STATE_ PROGRAM_STATEMENT_NUMBER	USTPGMSN	DECIMAL(10,0) Nullable	The statement number of the most recent user state program.
JOB_NAME	JOB_NAME	VARCHAR(10) Nullable	The job name of the job in which the authority check was made.
JOB_USER	JOB_USER	VARCHAR(10) Nullable	The job user of the job in which the authority check was made.
JOB_NUMBER	JOBNBR	CHAR(6) Nullable	The job number of the job in which the authority check was made.
THREAD_ID	THREAD_ID	BIGINT Nullable	The thread ID of the currently running thread of the job in which the authority check was made.
CURRENT_USER	CURUSR	VARCHAR(10) Nullable	The current user associated with the thread of the job in which the authority check was made.
OBJECT_FILE_ID	OFILEID	BINARY(16) Nullable	The file ID of the path name.
OBJECT_ASP_NAME	OASP	VARCHAR(10) Nullable	The name of the auxiliary storage pool to which storage for the object in the path name is allocated.
OBJECT_ASP_NUMBER	OASPN	DECIMAL(5,0) Nullable	The number of the auxiliary storage pool to which storage for the object in the path name is allocated. A value of 0 indicates *SYSBAS.
PATH_NAME	PATH_NAME	DBCLOB(16M) CCSID 1200 Nullable	The path of the object whose authority was checked. This field contains information for document library objects (*DOC and *FLR object types), and objects in the "root" (/), QOpenSys, and user-defined file systems. This field will not be filled in for objects in libraries.
PATH_REGION	PATHREGION	CHAR(2) Nullable	The country or region id for the path name.
PATH_LANGUAGE	PATHLANG	CHAR(3) Nullable	The language id for the path name.
ABSOLUTE_PATH_INDICATOR	ABSPATHIND	CHAR(1) Nullable	Indicates whether the path name of the object is an absolute path or a relative path. This field is set to 'Y' if the path name of the object begins with a delimiter (path name resolution starts at the "root" (/) directory). This field is set to 'N' if the path name of the object contains a relative path name. In addition, when this field contains 'N', the **RELATIVE_DIRECTORY_FILE_ID** field contains the File ID of the parent directory of the relative path which is used to form an absolute path name.
RELATIVE_DIRECTORY_FILE_ID	RELDIRID	BINARY(16) Nullable	The relative directory file ID of the parent directory that contains the object in the **PATH_NAME** field. This field is set when the **ABSOLUTE_PATH_INDICATOR** field is 'N'.

Authority field values

The **REQUIRED_AUTHORITY** field, **CURRENT_AUTHORITY** field, and **CURRENT_ADOPTED_AUTHORITY** field can contain one of the values listed below.

- *ALL - Allows all operations on the object except those that are limited to the owner or controlled by authorization list management authority. This value is made up of the following detailed authority values: *OBJEXIST, *OBJMGT, *OBJOPR, *OBJALTER, *OBJREF, *READ, *ADD, *DLT, *UPD, *EXECUTE.
- *CHANGE - Allows all operations on the object except those that are limited to the owner or controlled by object existence authority, object alter authority, object reference authority, and object management authority. This value is made up of the following detailed authority values: *OBJOPR, *READ, *ADD, *DLT, *UPD, *EXECUTE.
- *USE - Allows access to the object attributes and use of the object. The user cannot change the object. This value is made up of the following detailed authority values: *OBJOPR, *READ, *EXECUTE.
- *EXCLUDE - All operations on the object are prohibited.

Detailed authority field values

The **DETAILED_REQUIRED_AUTHORITY** field, **DETAILED_CURRENT_AUTHORITY** field, and **DETAILED_CURRENT_ADOPTED_AUTHORITY** field can contain one or more of the values listed below.

- *OBJALTER: Object alter - provides authority to change the attributes of an object, such as adding or removing triggers and adding members for a database file.
- *OBJEXIST: Object existence - provides authority to control the object's existence and ownership.
- *OBJMGT: Object management - provides authority to specify security, to move or rename the object, and to add members if the object is a database file.
- *OBJOPR: Object operational - provides authority to look at the object's attributes and to use the object as specified by the data authorities that the user has to the object.
- *OBJREF: Object reference -provides authority to specify the object as the first level in a referential constraint.
- *ADD: Add - provides authority to add entries to the object.
- *DLT: Delete - provides authority to remove entries from the object.
- *EXECUTE: Execute - provides authority to run a program or search a library or directory.
- *READ: Read - provides authority to access the contents of the object.
- *UPD: Update - provides authority to change the content of existing entries in the object.
- *EXCLUDE: Exclude - all operations on the object are prohibited.
- *AUTLMGT: Authorization list management – the authority required to add, change or remove users and their authority from an Authorization List object.
- *OWNER: Ownership – the user owns the object and has all object and data authorities.

Chapter 11. Code license and disclaimer information

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

SUBJECT TO ANY STATUTORY WARRANTIES WHICH CANNOT BE EXCLUDED, IBM, ITS PROGRAM DEVELOPERS AND SUPPLIERS MAKE NO WARRANTIES OR CONDITIONS EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, REGARDING THE PROGRAM OR TECHNICAL SUPPORT, IF ANY.

UNDER NO CIRCUMSTANCES IS IBM, ITS PROGRAM DEVELOPERS OR SUPPLIERS LIABLE FOR ANY OF THE FOLLOWING, EVEN IF INFORMED OF THEIR POSSIBILITY:

1. LOSS OF, OR DAMAGE TO, DATA;
2. DIRECT, SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES, OR FOR ANY ECONOMIC CONSEQUENTIAL DAMAGES; OR
3. LOST PROFITS, BUSINESS, REVENUE, GOODWILL, OR ANTICIPATED SAVINGS.

SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF DIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, SO SOME OR ALL OF THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

Appendix A. Security commands

This section contains the system commands related to security. You can use these commands in place of the system menus by typing these commands on a command line. The commands are divided into task-oriented groups.

The Control language (CL) topic contains more detailed information about these commands. The tables in Appendix D, "Authority required for objects used by commands," on page 371 show what object authorities are required to use these commands.

For more information about tools and suggestions about how to use the security tools, see the Configuring the system to use security tools topic.

Authority holders commands

This table provides a list of the commands that allow you to work with authority holders.

Table 136. Authority holders commands

Command name	Descriptive name	Function
CRTAUTHLR	Create Authority Holder	Secure a file before the file exists. Authority holders are valid only for program-described database files.
DLTAUTHLR	Delete Authority Holder	Delete an authority holder. If the associated file exists, the authority holder information is copied to the file.
DSPAUTHLR	Display Authority Holder	Display all the authority holders on the system.

Authority lists commands

You can use these commands to perform different tasks on authority lists.

Table 137. Authority lists commands

Command name	Descriptive name	Function
ADDAUTLE	Add Authorization List Entry	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	Change users' authorities to the objects on the authorization list.
CRTAUTL	Create Authorization List	Create an authorization list.
DLTAUTL	Delete Authorization List	Delete an entire authorization list.
DSPAUTL	Display Authorization List	Display a list of users and their authorities to an authorization list.
DSPAUTLOBJ	Display Authorization List Objects	Display a list of objects secured by an authorization list.
EDTAUTL	Edit Authorization List	Add, change, and remove users and their authorities on an authorization list.
RMVAUTLE	Remove Authorization List Entry	Remove a user from an authorization list.

Table 137. Authority lists commands (continued)		
Command name	**Descriptive name**	**Function**
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	Work with authorization lists from a list display.

Object authority and auditing commands

You can refer to this table for commands that you can use to work with object authority and auditing.

Table 138. Object authority and auditing commands		
Command name	**Descriptive name**	**Function**
CHGAUD	Change Auditing	Change the auditing value for an object.
CHGAUT	Change Authority	Change the authority of users to objects.
CHGOBJAUD	Change Object Auditing	Specify whether access to an object is audited.
CHGOBJOWN	Change Object Owner	Change the ownership of an object from one user to another.
CHGOBJPGP	Change Object Primary Group	Change the primary group for an object to another user or to no primary group.
CHGOWN	Change Owner	Change the ownership of an object from one user to another.
CHGPGP	Change Primary Group	Change the primary group for an object to another user or to no primary group.
DSPAUT	Display Authority	Display users' authority to an object.
DSPLNK	Display Links	Show a list of names of specified objects in directories and options to display information about the objects.
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	Add, change, or remove a user's authority for an object.
GRTOBJAUT	Grant Object Authority	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	Remove one or more (or all) of the authorities given specifically to a user for the named objects.
WRKAUT	Work with Authority	Work with object authority by selecting options on a list display.

Table 138. Object authority and auditing commands (continued)		
Command name	**Descriptive name**	**Function**
WRKLNK	Work with Links	Show a list of names of specified objects in directories and options to work with the objects.
WRKOBJ	Work with Objects	Work with object authority by selecting options on a list display.
WRKOBJOWN	Work with Objects by Owner	Work with the objects owned by a user profile.
WRKOBJPGP	Work with Objects by Primary Group	Work with the objects for which a profile is the primary group using options from a list display.
WRKOBJPVT	Work with Objects by Private Authorities	Work with the objects for which a profile is privately authorized, using options from a list display.

Passwords commands

These commands enable the security administrator to assign, change, verify, or reset password associated with a user profile.

Table 139. Passwords commands		
Command name	**Descriptive name**	**Function**
CHGDSTPWD	Change Dedicated Service Tools Password	Reset the DST security capabilities profile to the default password shipped with the system.
CHGPWD	Change Password	Change the user's own password.
CHGUSRPRF	Change User Profile	Change the values specified in a user's profile, including the user's password.
CHKPWD	Check Password	Verify 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 cannot specify that the *USRPRF is to be created into an independent auxiliary storage pool (ASP). However, when a user is privately authorized to an object on an independent ASP, the user is the owner of an object on an independent ASP, or the user is the primary group of an object on an independent ASP, the profile's name is stored on the independent ASP. If the independent ASP 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 profiles commands

As a security administrator, you will need to use these commands to work with user profiles.

Table 140. User profiles commands

Command name	Descriptive name	Function
CHGPRF	Change Profile	Change some of the attributes of the user's own profile.
CHGUSRAUD	Change User Audit	Specify the action and object auditing for a user profile.
CHGUSRPRF	Change User Profile	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 path name to ensure the objects have not been tampered with.
CRTUSRPRF	Create User Profile	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	Delete a user profile from the system. This command provides an option to delete or change ownership of objects owned by the user profile.
DMPUSRPRF	Dump User Profile	Allows you to dump the user profile and related information.
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 an IBM i NetServer LAN manager password, the date the password was last changed, and the user profile text.
DSPSSTUSR	Display Service Tools User ID	Displays a list of service tools user identifiers. It can also be used to show detailed information about a specific service tools user ID, including the status and privileges of that user.
DSPUSRPRF	Display User Profile command	Display a user profile in several different formats.
GRTUSRAUT	Grant User Authority	Copy private authorities from one user profile to another user profile.
PRTPRFINT	Print Profile Internals	Print a report of internal information about the number of entries.
PRTUSRPRF	Print User Profile	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	Work with user profiles by entering options on a list display.

Related user profile commands

This table lists some other commands that are related to user profiles. These commands allow you to restore or save the user profiles and their attributes.

Table 141. Related user profile commands

Command name	Descriptive name	Function
DSPPGMADP	Display Programs That Adopt	Display a list of programs and SQL packages that adopt a specified user profile.
RSTAUT	Restore Authority	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	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.

Auditing commands

You can use these commands to manage auditing on an object.

Table 142. Auditing commands

Command name	Descriptive name	Function
CHGAUD	Change Auditing	Specify the auditing for an object.
CHGDLOAUD	Change Document Library Object Auditing	Specify whether access is audited for a document library object.
CHGOBJAUD	Change Object Auditing	Specify the auditing for an object.
CHGUSRAUD	Change User Audit	Specify the action and object auditing for a user profile.

Document library objects commands

This table lists the commands that you can use to work with document library objects.

Table 143. Document library objects commands

Command name	Descriptive name	Function
ADDDLOAUT	Add Document Library Object Authority	Give a user access to a document or folder or to secure a document or folder with an authorization list or an access code.

Table 143. Document library objects commands (continued)		
Command name	**Descriptive name**	**Function**
CHGDLOAUD	Change Document Library Object Auditing	Specify the object auditing level for a document library object.
CHGDLOAUT	Change Document Library Object Authority	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	Change the primary group for a document library object.
DSPAUTLDLO	Display Authorization List Document Library Objects	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	Display authority information for a document or a folder.
EDTDLOAUT	Edit Document Library Object Authority	Add, change, or remove users' authorities to a document or folder.
GRTUSRPMN	Grant User Permission	Gives permission to a user to handle documents and folders or to do office-related tasks on behalf of another user.
RMVDLOAUT	Remove Document Library Object Authority	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.

Server authentication entries commands

These commands allow you to display, add, remove, or change server authentication entries for a user profile.

Table 144. Server authentication entries commands		
Command name	**Descriptive name**	**Function**
ADDSVRAUTE	Add Server Authentication Entry	Add server authentication information for a user profile.
CHGSVRAUTE	Change Server Authentication Entry	Change existing server authentication entries for a user profile.
DSPSVRAUTE	Display Server Authentication Entries	Display server authentication entries for a user profile.
RMVSVRAUTE	Remove Server Authentication Entry	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.		

System distribution directory commands

You can use these commands to add, remove, change, rename, or display entries in the system distribution directory.

Table 145. System distribution directory commands

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.
DSPDIRE	Display Directory Entries	Display, print, or create a database file for some or all system distribution directory entries.
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.
RNMDIRE	Rename Directory Entry	Renames a local or remote user ID and address to a new user ID and address. This will rename all occurrences of the specified user ID and address in all IBM-supplied files.
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.

Validation lists commands

These two commands allow you to create and delete validation lists in a library.

Table 146. Validation lists commands

Command name	Descriptive name	Function
CRTVLDL	Create Validation List	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	Delete the specified validation list from a library.

Function usage information commands

You can use these commands to change or display function usage information.

Table 147. Function usage information commands

Command name	Descriptive name	Function
CHGFCNUSG	Change function usage	Change the usage information for a registered function.

Table 147. Function usage information commands (continued)		
Command name	**Descriptive name**	**Function**
DSPFCNUSG	Display function usage	Display a list of function identifiers and the detailed usage information for a specific function.
WRKFCNUSG	Work with function usage	Display a list of function identifiers and change or display function usage information.

Auditing security tools commands

These commands enable you to work with security auditing, the entries from the security audit journal and the system values that control security auditing.

For more information about the security tools, see Appendix G, "Commands and menus for security commands," on page 893.

Table 148. Auditing security tools commands		
Command name	**Descriptive name**	**Function**
CHGSECAUD	Change Security Auditing	Set up security auditing and to change the system values that control security auditing.
CPYAUDJRNE	Copy Audit Journal Entries	Copy entries from the security audit journal to output files that you can query. You can select specific entry types, specific users, and a time period.
DSPAUDJRNE[1]	Display Audit Journal Entries	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	Display information about the security audit journal and the system values that control security auditing.

1
IBM has stopped providing enhancements for the DSPAUDJRNE command. The command does not support all security audit record types, and the command does not list all the fields for the records it does support.

Authority security tools commands

You can use these commands to perform various printing tasks that are related to security settings.

Table 149. Authority security tools commands		
Command name	**Descriptive name**	**Function**
PRTJOBDAUT	Print Job Description Authority	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	Print a list of objects of the specified type whose public authority is not *EXCLUDE.
PRTPVTAUT	Print Private Authorities	Print a list of private authorities for objects of the specified type.

Table 149. Authority security tools commands (continued)		
Command name	**Descriptive name**	**Function**
PRTQAUT	Print Queue Authority	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	Print a list of subsystem descriptions in a library that contains a default user in a subsystem entry.
PRTTRGPGM	Print Trigger Programs	Print a list of trigger programs that are associated with database files on your system.
PRTUSROBJ	Print User Objects	Print a list of the user objects (objects not supplied by IBM) that are in a library.

System security tools commands

You can use these commands to work with system security.

Table 150. System security tools commands		
Command name	**Descriptive name**	**Function**
CHGSECA [1]	Change Security Attributes	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	Set security-relevant system values to their recommended settings. The command also sets up security auditing on your system.
CLRSVRSEC	Clear Server Security Data	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	Display the current and pending values of some system security attributes.
PRTCMNSEC	Print Communications Security	Print the security attributes of the *DEVD, *CTL, and *LIND objects on the system.
PRTSYSSECA	Print System Security Attributes	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	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.		

Appendix B. IBM-supplied user profiles

This section 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.

Related concepts

IBM-supplied user profiles

You can perform auditing tasks on IBM-supplied user profiles by verifying their passwords.

Default values for user profiles

This table 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 151. Default values for user profiles

User profile parameter	Default values	
	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
Block password change (PWDCHGBLK)	*SYSVAL	*SYSVAL
Local password management (LCLPWDMGT)	*YES	*YES
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

Table 151. Default values for user profiles (continued)

User profile parameter	Default values	
	IBM-supplied user profiles	**Create user profile display**
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
Character identifier control (CHRIDCTL)	*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
EIM association (EIMASSOC)	*NOCHG	*NOCHG
User expiration date (USREXPDATE)	*NONE	*NONE
Authority (AUT)	*EXCLUDE	*EXCLUDE
Action auditing (AUDLVL) [3]	*NONE	*NONE
Object auditing (OBJAUD) [3]	*NONE	*NONE

Table 151. Default values for user profiles (continued)		
User profile parameter	**Default values**	
	IBM-supplied user profiles	**Create user profile display**

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 cannot 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, the user is the owner of an object in an independent disk pool, or the user 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.

IBM-supplied user profiles

This table 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:

IBM-supplied user profiles 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 IBM i product and poses a security exposure until it is changed. However, Do not change any other values for IBM-supplied user profiles. Changing these profiles can cause system functions to fail.

- Authorities for IBM-supplied profiles

  Use caution when removing authorities that IBM-supplied profiles have for 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 can cause system functions to fail.

Table 152. IBM-supplied user profiles		
Profile name	**Descriptive name**	**Parameters different from default values**
QADSM	ADSM user profile	• USRCLS: *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
QANZAGENT	Trace Analyzer Agent Server	
QAUTPROF	IBM authority user profile	• CCSID: *HEX • SRTSEQ: *HEX
QBRMS	BRM user profile	• CCSID: *HEX • SRTSEQ: *HEX
QCLUMGT	Cluster management profile	• STATUS: *DISABLED • SPCENV: *NONE • MSGQ: *NONE • ATNPGM: *NONE
QCLUSTER	High availability cluster profile	• SPCAUT: *IOSYSCFG • SPCENV: *NONE • JOBD: QSYS/QCSTJOBD
QCOLSRV	Management central collection services user profile	
QDBSHR	Database share profile	• AUT: *ADD, *DELETE
QDBSHRDO	Database share profile	• AUT: *ADD, *DELETE
QDFTOWN	Default owner profile	• PTYLMT: 3

Table 152. IBM-supplied user profiles (continued)

Profile name	Descriptive name	Parameters different from default values
QDIRSRV	IBM i Directory Server server user profile	• LMTCPB: *YES • JOBD: QGPL/QBATCH • DSPSGNINF: *NO • LMTDEVSSN: *NO • DLVRY: *HOLD • SPCENV: *NONE • ATNPGM: *NONE
QDLFM	DataLink File Manager profile	• SRTSEQ: *HEX
QDOC	Document profile	• AUT: *CHANGE
QDSNX	Distributed systems node executive profile	• PTYLMT: 3 • CCSID: *HEX • SRTSEQ: *HEX
QEJBSVR	WebSphere® Application Server user profile	• SPCENV: *NONE
QEJB	Enterprise Java user profile	
QFNC	Finance profile	• PTYLMT: 3
QGATE	VM/MVS bridge profile	• CCSID: *HEX • SRTSEQ: *HEX
QIBMHELP	IBM Eclipse Online Help	
QIPP	Internet printing profile	
QLPAUTO	Licensed program automatic install profile	• USRCLS: *SYSOPR • INLMNU: *SIGNOFF • SPCAUT: *ALLOBJ, *JOBCTL ,*SAVSYS, *SECADM, *IOSYSCFG • INLPGM: QSYS/QLPINATO • DLVRY: *HOLD • SEV: 99
QLPINSTALL	Licensed program install profile	• USRCLS: *SYSOPR • DLVRY: *HOLD • SPCAUT: *ALLOBJ, *JOBCTL, *SAVSYS, *SECADM, *IOSYSCFG
QLWISVR	Default profile for IAS servers	• LMTDEVSSN: *NO • DSPSGNINF: *NO • LOCALE: *SYSVAL

Table 152. IBM-supplied user profiles (continued)		
Profile name	**Descriptive name**	**Parameters different from default values**
QMGTC	Management central profile	• SPCENV: *NONE • DSPSGNINF: *NO • LMTDEVSSN: *NO • JOBD: QSYS/QYPSJOBD
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	• CCSID: *HEX • SRTSEQ: *HEX
QNETSPLF	Network spooling profile	
QNTP	Network time profile	• JOBD: QTOTNTP • JOBD LIBRARY: QSYS
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
QPM400	IBM Performance Management for IBM i(PM IBM i)	• SPCAUT: *IOSYSCFG, *JOBCTL
QRDARSADM	Content Manager OnDemand user profile	• INLMNU: *SIGNOFF • TEXT: OnDemand Administration Profile
QRDAR	Content Manager OnDemand owning profile	• USRCLS: *PGMR • INLMNU: *SIGNOFF • OUTQ: *DEV • TEXT: OnDemand owning profile

Table 152. IBM-supplied user profiles (continued)

Profile name	Descriptive name	Parameters different from default values
QRDARS4001	Content Manager OnDemand owning profile 1	• INLMNU: *SIGNOFF • GRPPRF: QRDARS400 • OUTQ: *DEV • TEXT: OnDemand file owning profile 1
QRDARS4002	Content Manager OnDemand owning profile 2	• INLMNU: *SIGNOFF • GRPPRF: QRDARS400 • OUTQ: *DEV • TEXT: OnDemand file owning profile 2
QRDARS4003	Content Manager OnDemand owning profile 3	• INLMNU: *SIGNOFF • GRPPRF: QRDARS400 • OUTQ: *DEV • TEXT: OnDemand file owning profile 3
QRDARS4004	Content Manager OnDemand owning profile 4	• INLMNU: *SIGNOFF • GRPPRF: QRDARS400 • OUTQ: *DEV • TEXT: OnDemand file owning profile 4
QRDARS4005	Content Manager OnDemand owning profile 5	• INLMNU: *SIGNOFF • GRPPRF: QRDARS400 • OUTQ: *DEV • TEXT: OnDemand file 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 • UID: 0 • PASSWORD: QSECOFR
QSNADS	SNA distribution services profile	• CCSID: *HEX • SRTSEQ: *HEX
QSOC	OptiConnect user profile	• USRCLS: *SYSOPR • CURLIB: *QSOC • SPCAUT: *JOBCTL
QSPL	Spool profile	

Table 152. IBM-supplied user profiles (continued)		
Profile name	**Descriptive name**	**Parameters different from default values**
QSPLJOB	Spool job profile	
QSRV	Service profile	• USRCLS: *PGMR • SPCAUT: *ALLOBJ [1], *SAVSYS [1], *JOBCTL, *SERVICE • ASTLVL: *INTERMED • ATNPGM: QSYS/QSCATTN
QSRVAGT	Service Agent user profile	• USRCLS: *SYSOPR • SPCAUT: *ALLOBJ [1], *SAVSYS [1], *JOBCTL, *SERVICE, *IOSYSCFG • SPCENV: *NONE • DSPSGNINF: *NO • LMTDEVSSN: *NO • OUTQ: QSRVAGT/QS9SRVAGT
QSRVBAS	Service basic profile	• USRCLS: *PGMR • SPCAUT: *ALLOBJ [1] *SAVSYS [1] *JOBCTL • ASTLVL: *INTERMED • ATNPGM: QSYS/QSCATTN
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

Table 152. IBM-supplied user profiles (continued)

Profile name	Descriptive name	Parameters different from default values
QSYSOPR	System operator profile	• USRCLS: *SYSOPR • SPCAUT: *ALLOBJ [1], *SAVSYS, *JOBCTL • INLMNU: *LIBL/SYSTEM • MSGQ: QSYS/QSYSOPR • DLVRY: *BREAK • SEV: 40
QTCM	Triggered cache manager profile	• STATUS: *DISABLED • SPCENV: *NONE
QTCP	Transmission control protocol (TCP) profile	• USRCLS: *SYSOPR • SPCAUT: *JOBCTL • CCSID: *HEX • SRTSEQ: *HEX
QTFTP	Trivial File Transfer Protocol	• CCSID: *HEX • SRTSEQ: *HEX
QTMPLPD	Remote LPR user profile	• PTYLMT: 3 • JOBD: QGPL/QDFTJOBD • PWDEXPITV: *NOMAX • MSGQ: QTCP/QTMPLPD • AUT: *OBJOPR
QTMTWSG	HTML Workstation Gateway Profile user profile	• TEXT: HTML Workstation Gateway Profile
QTMHHTTP	HTML Workstation Gateway Profile user profile	• TEXT: HTTP Server Profile
QTMHHTP1	HTML Workstation Gateway Profile user profile	• MSGQ: QUSRSYS/QTMHHTTP • TEXT: HTTP Server CGI Profile
QTSTRQS	Test request profile	
QUSER	Workstation user profile	• PTYLMT: 3
QWEBADMIN	Profile for the Web Admin GUI	• LMTDEVSSN: *NO • DSPSGNINF: *NO
QWSERVICE	Default profile for Integrated Web Services server	• LMTDEVSSN: *NO • DSPSGNINF: *NO • LOCALE: *SYSVAL
QYCMCIMOM	Server user profile	

Table 152. IBM-supplied user profiles (continued)

Profile name	Descriptive name	Parameters different from default values
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

This section identifies which commands have restricted authorization (public authority is *EXCLUDE) when your system is shipped. It shows which 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 133.

In Table 153 on page 355, 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 371.

If you choose to grant other users or the public *USE authority to these commands, update this table to indicate that which commands are no longer restricted on your system. Using some commands might 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 371 for the object authorities required for commands.

Note: Proxy Commands

- The commands listed in table Table 153 on page 355 are shipped by IBM with public authority of *EXCLUDE. If you notice on your system that the public authority of a command listed in this table shows a value of *USE and is in the QSYS library then this command might be a proxy command. Proxy commands are shipped by IBM in the QSYS library and have public authority of *USE. It is the actual command in the product library that will have public authority of *EXCLUDE (unless the public authority has been changed by an authorized user of your system). Authority to the proxy command is checked by the system as well as authority to the actual target command in the product library.
- To determine if the command is a proxy command, run the **DSPOBJD** command specifying the command name in library QSYS on the **DSPOBJD OBJ** parameter. If the command is a proxy command it will show an attribute value of PRX. To determine the authority of the actual target command in the product library, first use the **DSPCMD** command on the proxy command in the QSYS library. This will show you the current proxy chain. Then, use the **DSPOBJAUT** command and specify the library qualified command name of the last target command in the current proxy chain. This will show you the authority on the actual command in the product library.

Table 153. Authorities of IBM-supplied user profiles to restricted commands

Command Name	QPGMR	QSYSOPR	QSRV	QSRVBAS
ADDASPCPYD				
ADDCADMRE				
ADDCADNODE				
ADDCLUMON				
ADDCLUNODE				
ADDCMDCRQA	S	S	S	S
ADDCRGDEVE				
ADDCRGNODE				

Table 153. Authorities of IBM-supplied user profiles to restricted commands (continued)

Command Name	QPGMR	QSYSOPR	QSRV	QSRVBAS
ADDCRSDMNK				
ADDDEVDMNE				
ADDDIRINST				
ADDDSTQ	S	S		
ADDDSTRTE	S	S		
ADDDSTSYSN	S	S		
ADDEXITPGM				
ADDDWDFN				
ADDHACFGD				
ADDHAPCY				
ADDHYSSTGD				
ADDJWDFN				
ADDMFS				
ADDMSTPART				
ADDNETJOBE				
ADDOBJCRQA	S	S	S	S
ADDOPTCTG				
ADDOPTSVR				
ADDPEXDFN	S		S	
ADDPEXFTR	S		S	
ADDPRDCRQA	S	S	S	S
ADDPTFCRQA	S	S	S	S
ADDRPYLE	S			
ADDRSCCRQA	S	S	S	S
ADDSVCCPYD				
ADDTRCFTR				
ADDWLCGRP				
ADDWLCPRDE				
ANSQST				
ANZCMDPFR				
ANZDBF				
ANZDBFKEY				
ANZDFTPWD				
ANZJVM	S	S	S	S

Table 153. Authorities of IBM-supplied user profiles to restricted commands (continued)

Command Name	QPGMR	QSYSOPR	QSRV	QSRVBAS
ANZOBJCVN				
ANZPFRDTA				
ANZPGM				
ANZPRB	S	S	S	S
ANZPRFACT				
ANZS34OCL				
ANZS36OCL				
APYJRNCHG	S		S	
APYPTF			S	
APYRMTPTF	S	S	S	S
DSPHACFGD				
CFGACCWEB				
CFGCRGCNR				
CFGDEVASP				
CFGDSTSRV	S	S		
CFGGEOMIR				
CFGRPDS	S	S		
CFGSYSSEC				
CHGACTSCDE				
CHGASPA				
CHGASPACT				
CHGASPCPYD				
CHGASPSSN				
CHGAUTCOL				
CHGCAD				
CHGCLU				
CHGCLUCFG				
CHGCLUMON				
CHGCLUNODE				
CHGCLURCY				
CHGCLUVER				
CHGCMDCRQA	S	S	S	S
CHGCRG				
CHGCRGCNR				

Table 153. Authorities of IBM-supplied user profiles to restricted commands (continued)

Command Name	QPGMR	QSYSOPR	QSRV	QSRVBAS
CHGCRGDEVE				
CHGCRGPRI				
CHGCRSDMNK				
CHGCSMSSN				
CHGDIRSRVA				
CHGDSTQ	S	S		
CHGDSTRTE	S	S		
CHGEXPSCDE				
CHGFCNARA				
CHGGPHFMT				
CHGGPHPKG				
CHGHACFGD				
CHGHAPCY				
CHGHYSSTGD				
CHGHYSSTS				
CHGJOBTRC				
CHGJOBTYP				
CHGJRN	S	S	S	
CHGJRNA	S	S		
CHGLICINF				
CHGMGDSYSA	S	S	S	S
CHGMGRSRVA	S	S	S	S
CHGMSTK				
CHGNETA				
CHGNETJOBE				
CHGNFSEXP				
CHGNWSA				
CHGNWSCFG				
CHGOBJCRQA	S	S	S	S
CHGOPTA				
CHGPEXDFN	S		S	
CHGPRB	S	S	S	S
CHGPRDCRQA	S	S	S	S
CHGPTFCRQA	S	S	S	S

Table 153. Authorities of IBM-supplied user profiles to restricted commands (continued)				
Command Name	**QPGMR**	**QSYSOPR**	**QSRV**	**QSRVBAS**
CHGPTR			S	
CHGQSTDB				
CHGRCYAP	S	S		
CHGRPYLE	S			
CHGRSCCRQA	S	S	S	S
CHGSVCCPYD				
CHGSVCSSN				
CHGSYSLIBL				
CHGSYSVAL	S	S	S	
CHGS34LIBM				
CHGWLCGRP				
CHKASPBAL				
CHKCMNTRC			S	
CHKMSTKVV				
CHKPRDOPT	S	S	S	S
CLRMSTKEY				
CPHDTA				
CPYFCNARA				
CPYFRMLDIF				
CPYFRMMSD				
CPYGPHFMT				
CPYGPHPKG				
CPYPFRCOL				
CPYPFRDTA				
CPYPTF	S	S	S	S
CPYPTFGRP	S	S	S	S
CPYTOLDIF				
CPYTOMSD				
CRTADMDMN				
CRTAUTHLR				
CRTCAD				
CRTCLS				
CRTCLS				
CRTCLU				

Table 153. Authorities of IBM-supplied user profiles to restricted commands (continued)				
Command Name	**QPGMR**	**QSYSOPR**	**QSRV**	**QSRVBAS**
CRTCRG				
CRTCRGCNR				
CRTFCNARA				
CRTGPHFMT				
CRTGPHPKG				
CRTHSTDTA				
CRTJOBD				
CRTNWSCFG				
CRTPFRDTA				
CRTPFRSUM				
CRTLASREP	S			
CRTPEXDTA	S		S	
CRTQSTDB				
CRTQSTLOD				
CRTSBSD	S	S		
CRTUDFS				
CRTUDFS				
CRTVLDL				
CVTBASSTR				
CVTBASUNF				
CVTBGUDTA				
CVTDIR				
CVTPFRCOL				
CVTPFRDTA				
CVTPFRTHD				
CVTS36FCT				
CVTS36JOB				
CVTS38JOB				
CVTTCPCL	S	S	S	S
DB2LDIF				
DLTADMDMN				
DLTAPARDTA	S	S	S	S
DLTAUTCOL				
DLTCAD				

Table 153. Authorities of IBM-supplied user profiles to restricted commands (continued)

Command Name	QPGMR	QSYSOPR	QSRV	QSRVBAS
DLTCLU				
DLTCMNTRC			S	
DLTCRGCLU				
DLTCRGCNR				
DLTEXPSPLF				
DLTFCNARA				
DLTGPHFMT				
DLTGPHPKG				
DLTHSTDTA				
DLTINTSVR				
DLTLICPGM				
DLTNWSCFG				
DLTPEXDTA	S		S	
DLTPFRCOL				
DLTPFRDTA				
DLTPRB	S	S	S	S
DLTPTF	S	S	S	S
DLTQST				
DLTQSTDB				
DLTRMTPTF	S	S	S	S
DLTSMGOBJ	S	S	S	S
DLTUDFS				
DLTVLDL				
DLTWNTSVR				
DMPDLO	S	S	S	S
DMPJOB	S	S	S	S
DMPJOBINT	S	S	S	S
DMPJVM	S	S	S	S
DMPMEMINF				
DMPOBJ			S	S
DMPSYSOBJ	S	S	S	S
DMPTRC	S		S	
DMPUSRPRF				
DSPASPCPYD				

Table 153. Authorities of IBM-supplied user profiles to restricted commands (continued)

Command Name	QPGMR	QSYSOPR	QSRV	QSRVBAS
DSPASPSSN				
DSPCLUINF				
DSPCRGCNR				
DSPCRGINF				
DSPCSMSSN				
DSPDSTLOG				
DSPHACFGD				
DSPHAPCY				
DSPHSTGPH				
DSPHYSSTGD				
DSPHYSSTS				
DSPMGDSYSA	S	S	S	S
DSPNWSCFG				
DSPPFRDTA				
DSPPFRGPH				
DSPPTF	S	S	S	S
DSPPTFAPYI	S	S	S	S
DSPPTFGRP	S	S	S	S
DSPSRVSTS	S	S	S	S
DSPSVCCPYD				
DSPSVCSSN				
DSPUSGINF				
DSPWLCGRP				
EDTCPCST		S		
EDTQST				
EDTRBDAP		S		
EDTRCYAP	S	S		
ENCCPHK				
ENCFRMMSTK				
ENCTOMSTK				
ENDACCWEB				
ENDASPBAL				
ENDASPSSN				
ENDAUTCOL				

Table 153. Authorities of IBM-supplied user profiles to restricted commands (continued)

Command Name	QPGMR	QSYSOPR	QSRV	QSRVBAS
ENDCAD				
ENDCHTSVR				
ENDCLUNOD				
ENDCMNTRC			S	
ENDCRG				
ENDCRGCNR				
ENDCSMSSN				
ENDDBGSVR	S	S	S	S
ENDDW				
ENDHOSTSVR	S	S	S	S
ENDIDXMON				
ENDJOBABN	S	S	S	
ENDJOBTRC				
ENDJW				
ENDMGDSYS	S	S	S	S
ENDMGRSRV	S	S	S	S
ENDMSF		S	S	S
ENDNFSSVR		S	S	S
ENDPEX	S		S	
ENDPFRTRC			S	
ENDSRVJOB	S	S	S	S
ENDSVCSSN				
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
ENDWCH				
GENCPHK				
GENCRSDMNK				
GENMAC				
GENPIN				
GENS36RPT				
GENS38RPT				

Table 153. Authorities of IBM-supplied user profiles to restricted commands (continued)				
Command Name	**QPGMR**	**QSYSOPR**	**QSRV**	**QSRVBAS**
GRTACCAUT				
HLDCMNDEV	S	S	S	S
HLDDSTQ	S	S		
INSINTSVR				
INSPTF[2]			S	
INSRMTPRD	S	S	S	S
INSWNTSVR				
INZDSTQ	S	S		
INZNWSCFG				
INZSYS				
LDIF2DB				
LODOPTFMW				
LODPTF			S	
LODQSTDB				
MGRS36				
MGRS36APF				
MGRS36CBL				
MGRS36DFU				
MGRS36DSPF				
MGRS36ITM				
MGRS36LIB				
MGRS36MNU				
MGRS36MSGF				
MGRS36QRY				
MGRS36RPG				
MGRS36SEC				
MGRS38OBJ				
MIGRATE				
MOVPFRCOL				
PKGPRDDST	S	S	S	S
PRTACTRPT				
PRTCMNTRC			S	
PRTCPTRPT				
PRTJOBRPT				

Table 153. Authorities of IBM-supplied user profiles to restricted commands (continued)				
Command Name	**QPGMR**	**QSYSOPR**	**QSRV**	**QSRVBAS**
PRTJOBTRC				
PRTLCKRPT				
PRTPOLRPT				
PRTRSCRPT				
PRTSYSRPT				
PRTTNSRPT				
PRTTRCRPT				
PRTDSKINF				
PRTERRLOG	S	S	S	S
PRTINTDTA	S	S	S	S
PRTPRFINT				
PWRDWNSYS		S		
RCLAPPN				
RCLDBXREF				
RCLOBJOWN				
RCLOPT				
RCLSPLSTG	S	S	S	S
RCLSTG	S	S	S	S
RCLTMPSTG	S	S	S	S
RESMGRNAM	S	S	S	S
RLSCMNDEV	S	S	S	S
RLSDSTQ	S	S		
RLSIFSLCK				
RLSRMTPHS	S	S		
RMVACC				
RMVACCWEB				
RMVASPCPYD				
RMVCADMRE				
RMVCADNODE				
RMVCLUMON				
RMVCLUNODE				
RMVCRGDEVE				
RMVCRGNODE				
RMVCRSDMNK				

Table 153. Authorities of IBM-supplied user profiles to restricted commands (continued)

Command Name	QPGMR	QSYSOPR	QSRV	QSRVBAS
RMVDEVDMNE				
RMVDFRID				
RMVDIRINST				
RMVDSTQ	S	S		
RMVDSTRTE	S	S		
RMVDSTSYSN	S	S		
RMVDWDFN				
RMVEXITPGM				
RMVHACFGD				
RMVHAPCY				
RMVHYSSTGD				
RMVJRNCHG	S		S	
RMVJWDFN				
RMVLANADP				
RMVMFS				
RMVNETJOBE				
RMVOPTCTG				
RMVOPTSVR				
RMVPEXDFN	S		S	
RMVPEXFTR	S		S	
RMVPTF			S	
RMVRMTPTF	S	S	S	S
RMVRPYLE	S			
RMVSVCCPYD				
RMVTRCFTR				
RMVWLCGRP				
RMVWLCPRDE				
RSTAUT				
RST[3]				
RSTCFG				
RSTDFROBJ				
RSTDLO				
RSTHAPCY				
RSTLIB				

Table 153. Authorities of IBM-supplied user profiles to restricted commands (continued)

Command Name	QPGMR	QSYSOPR	QSRV	QSRVBAS
RSTLICPGM				
RSTOBJ[3]				
RSTPFRCOL				
RSTPFRDTA				
RSTS36F				
RSTS36FLR				
RSTS36LIBM				
RSTS38AUT				
RSTUSRPRF				
RTVCSMSSN				
RTVSVCCPYD				
RTVSVCSSN				
RTVDSKINF				
RTVPRD	S	S	S	S
RTVPTF	S	S	S	S
RTVSMGOBJ	S	S	S	S
RTVTCPINF				
RUNLPDA	S	S	S	S
RUNSMGCMD	S	S	S	S
RUNSMGOBJ	S	S	S	S
RVKPUBAUT				
SAVAPARDTA	S	S	S	S
SAVHAPCY				
SAVLICPGM				
SAVPFRCOL				
SAVPFRDTA				
SAVRSTCHG				
SAVRSTLIB				
SAVRSTOBJ				
SBMFNCJOB				
SBMNWSCMD				
SETMSTK				
SETMSTKEY				
SNDDSTQ	S	S		

Table 153. Authorities of IBM-supplied user profiles to restricted commands (continued)				
Command Name	**QPGMR**	**QSYSOPR**	**QSRV**	**QSRVBAS**
SNDPRD	S	S	S	S
SNDPTF	S	S	S	S
SNDPTFORD			S	S
SNDSMGOBJ	S	S	S	S
SNDSRVRQS			S	S
STRACCWEB				
STRASPBAL				
STRASPSSN				
STRAUTCOL				
STRCAD				
STRCHTSVR				
STRCLUNOD				
STRCMNTRC			S	
STRCRG				
STRCRGCNR				
STRCSMSSN				
STRDBG	S		S	S
STRDBGSVR	S	S	S	S
STRDW				
STRHOSTSVR	S	S	S	S
STRIDXMON				
STRJW				
STRJOBTRC				
STRMGDSYS	S	S	S	S
STRMGRSRV	S	S	S	S
STRMSF[1]		S	S	S
STRNFSSVR				
STRNETINS				
STROBJCVN				
STRPEX	S		S	
STRPFRG				
STRPFRT				
STRPFRTRC			S	
STRRGZIDX				

Table 153. Authorities of IBM-supplied user profiles to restricted commands (continued)				
Command Name	**QPGMR**	**QSYSOPR**	**QSRV**	**QSRVBAS**
STRSPLRCL				
STRSRVJOB	S	S	S	S
STRSST			S	
STRSVCSSN				
STRSYSMGR	S	S	S	S
STRS36MGR				
STRS38MGR				
STRTCP	S	S	S	S
STRTCPIFC	S	S	S	S
STRTCPSVR	S	S	S	S
STRUPDIDX				
STRWCH				
TRCASPBAL				
TRCCPIC				
TRCICF				
TRCINT	S		S	
TRCJOB	S	S	S	S
TRCTCPAPP			S	S
TRNPIN				
UPDPTFINF				
UPDTCPINF				
VFYCMN	S	S	S	S
VFYLNKLPDA	S	S	S	S
VFYMSTK				
VFYPIN				
VFYPRT	S	S	S	S
VFYTAP	S	S	S	S
WRKASPCPYD				
WRKCADMRE				
WRKCNTINF			S	S
WRKDEVTBL				
WRKDPCQ	S	S		
WRKDSTQ	S	S		
WRKFCNARA				

Table 153. Authorities of IBM-supplied user profiles to restricted commands (continued)

Command Name	QPGMR	QSYSOPR	QSRV	QSRVBAS
WRKHACFGD				
WRKHAPCY				
WRKHYSSTS				
WRKJRN	S	S	S	
WRKLICINF				
WRKNWSCFG				
WRKPEXDFN	S		S	
WRKPEXFTR	S		S	
WRKPGMTBL				
WRKPRB	S	S	S	S
WRKPTFGRP	S	S	S	S
WRKPTFORD			S	S
WRKSRVPVD			S	S
WRKSYSACT				
WRKTRC				
WRKTXTIDX				
WRKUSRTBL				
WRKWCH				

1 The QMSF user profile is also authorized to this command.

2 QSRV can only run this command if an IPL is not being done.

3 In addition to QSYS, user profile QRDARS400 has authority.

Appendix D. Authority required for objects used by commands

The tables in this section show what authority is needed for objects referenced by commands.

For example, in the entry for the Change User Profile (CHGUSRPRF) command the table lists all of the objects to which you need authority, 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 IBM i 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.

The following sections 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.

Authority required for object

The authorities specified in the tables show the object authorities and the data authorities that are required for the object when using the command.

Authority required 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 requires *READ and *ADD authority.

Object type

The value refers to the type of the object specified in the Referenced object column.

File system

The value refers to the type of file system that the referenced object belongs to.

For the integrated file system in the IBM i operating system, refer to Integrated file system.

The following table 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.

Table 154. 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.

Table 154. Description of authority types (continued)		
Authority	**Name**	**Functions allowed**
*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.
*OBJREF	Object Reference	Specify a database file as the parent in a referential constraint. For example, assume that 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.
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.

In addition to these values, the *Authority needed* columns of the table might show system-defined subsets of these authorities. The following table shows the subsets of object authorities and data authorities.

Table 155. System-defined authority				
Authority	***ALL**	***CHANGE**	***USE**	***EXCLUDE**
Object Authorities				
*OBJOPR	X	X	X	
*OBJMGT	X			
*OBJEXIST	X			

Table 155. System-defined authority (continued)				
Authority	***ALL**	***CHANGE**	***USE**	***EXCLUDE**
*OBJALTER	X			
*OBJREF	X			
Data Authorities				
*READ	X	X	X	
*ADD	X	X		
*UPD	X	X		
*DLT	X	X		
*EXECUTE	X	X	X	

The following table shows additional authority subsets that are supported by the CHGAUT and WRKAUT commands.

Table 156. 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

Command usage assumptions

There are some default assumptions you need to consider before using any command.

1. *USE authority is required to use any 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 that is used by the command. These files and panel groups are shipped with public authority *USE.

General rules for object authorities on commands

This table shows the 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	*X	
	Directory when pattern is specified (* or ?)	*R	
Creating object in directory	Directories in path prefix	*X	
	Directory to contain new object	*WX	
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	

Command	Referenced object	Authority needed	
		For object	**For library**
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 and *REPLACE option specified and member did not previously exist	*OBJOPR, *OBJMGT or *OBJALTER, *ADD, *DLT	*ADD, *EXECUTE
	Output file, if file exists and new member is added and *ADD option specified and member did not previously exist	OBJOPR, *OBJMGT or *OBJALTER, *ADD	*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 or *OBJALTER, *ADD, *DLT	*EXECUTE
	Format file (QA*xxxxx*), if output file does not exist	*OBJOPR	
Display (DSP) using *PRINT or Work (WRK) using *PRINT	Object to be displayed	*USE	*EXECUTE
	Output queue[4]	*READ	*EXECUTE
	Printer file (QP*xxxxx* 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

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 that is 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 additional authority to the output queue. A user with *SPLCTL special authority does not need any additional authority to the output queue.

5

For device files, *OBJOPR authority is also required.

Command	Referenced object	Authority needed	
		For object	For library
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.			

Common commands for most objects

This table lists commands that can work on most objects in alphabetical order.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, "Commands shipped with public authority *EXCLUDE," on page 355 shows which IBM-supplied user profiles are authorized to the command. The security officer can grant *USE authority to others.

Table 157. Common commands for most objects

Command	Referenced object	Authority needed	
		For object	For library
ALCOBJ [1,2,11]	Object	*OBJOPR	*EXECUTE
ANZOBJCVN (Q) [20]			
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 [3,4,36]	Object	*OBJEXIST	*EXECUTE
	Object (if file, library, subsystem description)	*OBJOPR, *OBJEXIST	*EXECUTE
	Object (if *AUTL)	Ownership or *ALLOBJ	*EXECUTE
	Old user profile	*DLT	*EXECUTE
	New user profile	*ADD	*EXECUTE
	ASP Device (if specified)	*USE	

Table 157. Common commands for most objects (continued)

Command	Referenced object	Authority needed	
		For object	**For library**
CHGOBJPGP [3,36]	Object	*OBJEXIST	*EXECUTE
	Object (if file, library, subsystem description)	*OBJOPR, *OBJEXIST	*EXECUTE
	Object (if *AUTL)	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 *AUTL	*AUTLMGT	*USE, *ADD
	Object being copied, all other types	*OBJMGT, *USE	*USE
	CRTSAVF command (if the object is a save file)	*OBJOPR	
	ASP Device (if specified)	*USE	
DCPOBJ	Object	*USE	*EXECUTE
DLCOBJ [1,11]	Object	*OBJOPR	*EXECUTE
DLTOBJ [35]	Object	*OBJEXIST	*EXECUTE
	ASP Device (if specified)	*USE	
DMPOBJ (Q) [3]	Object	*OBJOPR, *READ	*EXECUTE
DMPSYSOBJ (Q)	Object	*OBJOPR, *READ	*EXECUTE
DSPOBJAUT [3]	Object (to see all authority information) [36]	*OBJMGT or *ALLOBJ special authority or ownership	*EXECUTE
	Output file	Refer to the general rules.	Refer to the general rules.
	ASP Device (if specified) [36]	*USE	
DSPOBJD [2, 28]	Output file	Refer to the general rules.	Refer to the general rules.
	Object	Some authority other than *EXCLUDE	*EXECUTE
	ASP Device (if specified)	*EXECUTE	

Table 157. Common commands for most objects (continued)

Command	Referenced object	Authority needed	
		For object	**For library**
EDTOBJAUT [3,5,6,15,36]	Object	*OBJMGT	*EXECUTE
	Object (if file)	*OBJOPR, *OBJMGT	*EXECUTE
	*AUTL, if used to secure object	Not *EXCLUDE	
	ASP Device (if specified)	*USE	
ENDSAVSYNC[10]			
GRTOBJAUT [3,5,6,15,36]	Object	*OBJMGT	*EXECUTE
	Object (if file)	*OBJOPR, *OBJMGT	*EXECUTE
	*AUTL, if used to secure object	Not *EXCLUDE	
	ASP Device (if specified)	*USE	
	Reference ASP Device (if specified)	*EXECUTE	
	Reference object	*OBJMGT or Ownership	*EXECUTE
MOVOBJ [3,7,12]	Object	*OBJMGT	
	Object (if *FILE)	*ADD, *DLT, *EXECUTE	
	Object (not *FILE),	*DLT, *EXECUTE	
	From-library		*CHANGE
	To-library		*READ, *ADD
	ASP Device (if specified)	*USE	
PRTADPOBJ [26](Q)			
PRTPUBAUT [26]			
PRTUSROBJ [26]			
PRTPVTAUT [26]			
RCLDBXREF			
RCLOBJOWN (Q)			
RCLSTG (Q)			
RCLTMPSTG (Q)	Object	*OBJMGT	*EXECUTE
RMVDFRID (Q) [10]			
RNMOBJ [3,11]	Object	*OBJMGT	*UPD, *EXECUTE
	Object, if *AUTL	*AUTLMGT	*EXECUTE
	Object (if *FILE)	*OBJOPR, *OBJMGT	*UPD, *EXECUTE
	ASP Device (if specified)	*USE	

Table 157. Common commands for most objects (continued)

Command	Referenced object	Authority needed	
		For object	**For library**
RSTDFROBJ (Q) [10]	QSYS/QPSRLDSP printer output, if OUTPUT(*PRINT) specified	*USE	*EXECUTE
	Output file, if specified	Refer to the general rules	Refer to the general rules
	QSYS/QASRRSTO field reference file for output file, if an output file is specified and does not exist	*USE	*EXECUTE
RSTOBJ (Q)[3,13, 31, 33]	Object, if it already exists in the library	*OBJEXIST [8]	*EXECUTE, *ADD
	Object, if it is *CFGL, *CNNL, *CTLD, *DEVD, *LIND, or *NWID	*CHANGE and *OBJMGT	*EXECUTE
	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 (Q)	Tape unit or optical unit	*USE	*EXECUTE
	Tape (QSYSTAP) file or diskette (QSYSDKT) file	*USE [8]	*EXECUTE
	Optical File (OPTFILE)[22]	*R	Not applicable
	Parent Directory of optical file (OPTFILE)[22]	*X	Not applicable
	Path prefix of OPTFILE[22]	*X	Not applicable
	Optical volume[24]	*USE	Not applicable
	QSYS/QPSRLDSP printer output, if OUTPUT(*PRINT) specified	*USE	*EXECUTE
	Output file, if specified	Refer to the general rules.	Refer to the general rules.
	QSYS/QASRRSTO field reference file for output file, if an output file is specified and does not exist	*USE	*EXECUTE
	ASP device description[25]	*USE	

Table 157. Common commands for most objects (continued)

Command	Referenced object	Authority needed	
		For object	**For library**
RSTSYSINF	Save file	*USE	*EXECUTE
	Tape unit or optical unit	*USE	*EXECUTE
	Optical File (OPTFILE)[22]	*R	Not applicable
	Parent Directory of optical file (OPTFILE)[22]	*X	Not applicable
	Path prefix of OPTFILE[22]	*X	Not applicable
	Optical volume[24]	*USE	Not applicable
RVKPUBAUT [20]			
RTVOBJD [2, 29]	Object	Some authority other than *EXCLUDE	*EXECUTE
RVKOBJAUT [3,5,15, 27,36]	ASP Device (if specified)	*USE	
SAVCHGOBJ [3, 32]	Object (8)	*OBJEXIST	*EXECUTE
	Tape unit or 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 user space, if specified	*USE	*EXECUTE
SAVCHGOBJ	Optical File (OPTFILE)[22]	*RW	Not applicable
	Parent Directory of optical file (OPTFILE)[22]	*WX	Not applicable
	Path prefix of optical file (OPTFILE)[22]	*X	Not applicable
	Root Directory (/) of optical volume[22, 23]	*RWX	Not applicable
	Optical volume[24]	*CHANGE	
	Output file, if specified	Refer to the general rules.	Refer to the general rules.
	QSYS/QASAVOBJ field reference file for output file, if an output file is specified and does not exist	*USE [8]	*EXECUTE
	QSYS/QPSAVOBJ printer output	*USE [8]	*EXECUTE
	ASP device description[25]	*USE	

Table 157. Common commands for most objects (continued)

Command	Referenced object	Authority needed	
		For object	**For library**
SAVOBJ [3, 32]	Object	*OBJEXIST [8]	*EXECUTE
	Media definition	*USE	*EXECUTE
	Tape unit or 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 user space, if specified	*USE	*EXECUTE
SAVOBJ	Optical File (OPTFILE) [22]	*RW	Not applicable
	Parent Directory of optical file (OPTFILE)[22]	*WX	Not applicable
	Path prefix of OPTFILE[22]	*X	Not applicable
	Root directory (/) of optical volume [22, 23]	*RWX	Not applicable
	Optical volume[24]	*CHANGE	
	Output file, if specified	Refer to the general rules.	Refer to the general rules.
	QSYS/QASAVOBJ field reference file for output file, if an output file is specified and does not exist	*USE [8]	*EXECUTE
	QSYS/QPSAVOBJ printer output	*USE [8]	*EXECUTE
	ASP device description[25]	*USE	
SAVSTG [10]			
SAVSYS [10]	Tape unit, optical unit	*USE	*EXECUTE
	Root directory (/) of optical volume[22]	*RWX	Not applicable
	Optical volume[24]	*CHANGE	Not applicable
SAVSYSINF	Media definition	*USE	*EXECUTE
	Tape unit or optical unit	*USE	*EXECUTE
	Save file, if empty	*USE, *ADD	*EXECUTE
	Save file, if records exist in it	*OBJMGT, *USE, *ADD	*EXECUTE
	Optical File (OPTFILE) [22]	*RW	Not applicable
	Parent Directory of optical file (OPTFILE)[22]	*WX	Not applicable
	Path prefix of OPTFILE[22]	*X	Not applicable
	Root directory (/) of optical volume [22, 23]	*RWX	Not applicable
	Optical volume[24]	*CHANGE	

Table 157. Common commands for most objects (continued)

Command	Referenced object	Authority needed	
		For object	**For library**
SAVRSTCHG	On the source system, same authority as required by SAVCHGOBJ command.		
	On the target system, same authority as required by RSTOBJ command.		
	ASP device description[25]	*USE	
SAVRSTOBJ	On the source system, same authority as required by SAVOBJ command.		
	On the target system, same authority as required by RSTOBJ command.		
	ASP device description[25]	*USE	
SETOBJACC	Object	*OBJOPR	*EXECUTE
STROBJCVN (Q)[20]			
STRSAVSYNC[34]			
WRKOBJ [19,36]	Object	Any authority	*USE
WRKOBJLCK	Object		*EXECUTE
	ASP Device	*EXECUTE	
WRKOBJOWN [17]	User profile	*READ	*EXECUTE
WRKOBJPGP [17]	User profile	*READ	*EXECUTE
WRKOBJPVT[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.

Table 157. Common commands for most objects (continued)

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 authority to the from-object. Authorities can be obtained from adopted authority or through the group profile. • If an error occurs while copying authorities to the new object, the newly created object is deleted. **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 a value other than *NONE for the Allow object differences (ALWOBJDIF) parameter. **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 do one of the following actions: • 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.			

Table 157. Common commands for most objects (continued)

Command	Referenced object	Authority needed	
		For object	For library
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 might 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. **25** Authority required only if save or restore operation requires a library namespace switch.			
26 You must have *ALLOBJ or *AUDIT special authority to use this command. **27** ***** Security Risk ***** Revoking all authorities specifically given to a user for an object can result in the user having more authority than before the revoke operation. If a user has *USE authority for and object and *CHANGE authority on the authorization list that secures the object, revoking *USE authority results in the user having *CHANGE authority to the object. **28** You must have either *ALLOBJ or *AUDIT special authority to have the current object auditing value displayed. Otherwise, the value *NOTAVL is displayed to indicate that the value is not available for display. **29** You must have either *ALLOBJ or *AUDIT special authority to retrieve the current object auditing value. Otherwise, the value *NOTAVL is returned to indicate that the values are not available for retrieval. **30** See the CHGPGM, CHGSRVPGM, and CHGMOD commands to determine the authority needed to convert programs, service programs, and modules.			
31 You must have *ALLOBJ special authority to specify *YES for the PVTAUT parameter. **32** You must have either *ALLOBJ or *SAVSYS special authority to specify *YES for the PVTAUT parameter. **33** You must have *SAVSYS special authority to specify a name for the DFRID parameter. **34** You must have *SAVSYS and *JOBCTL special authority. **35** Some supported object types may require additional object and library authorities. Refer to the Delete Object (QLIDLTO) API documentation for more information. **36** If you are authorized to the IBM i Database Security Administrator function (QIBM_DB_SECADM) you do not need the specified special authority or the specified authority to the object. However, users authorized to the QIBM_DB_SECADM function cannot grant authority to themselves or transfer ownership to themselves unless they have the authorities required for the operation.			

Access path recovery commands

This table lists the specific authorities required for the access path recovery commands

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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.

Command	Referenced object	Authority needed	
		For object	For library
CHGRCYAP [1] (Q)	ASP Device (if specified)	*USE	
DSPRCYAP [1]	ASP Device (if specified)	*USE	
EDTRBDAP [2] (Q)			
EDTRCYAP [1] (Q)	ASP Device (if specified)	*USE	

1
You must have *JOBCTL special authority to use this command.

2
You must have *ALLOBJ special authority to use this command.

IBM i Access for Web commands

This table lists the specific authorities required for the IBM i access for web commands

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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.

Command	Referenced object	Authority needed	
		For object	For library
CFGACCWEB [1] (Q)			
ENDACCWEB [1] (Q)			
RMVACCWEB [1] (Q)			
STRACCWEB [1] (Q)			

1
You must have *ALLOBJ special authority to use this command.

Advanced Function Presentation (AFP) commands

This table lists the specific authorities required for the Advanced Function Presentation (AFP) commands.

Command	Referenced object	Authority needed	
		For object	For library
ADDFNTTBLE	DBCS font table	*CHANGE	*EXECUTE
CHGCDEFNT	Font resource	*CHANGE	*EXECUTE
CHGFNTTBLE	DBCS font table	*CHANGE	*EXECUTE

Command	Referenced object	Authority needed	
		For object	For library
CRTFNTRSC	Source file	*USE	*EXECUTE
	Font resource: REPLACE(*NO)		*READ, *ADD
	Font resource: REPLACE(*YES)	Refer to the general rules.	*READ, *ADD
CRTFNTTBL	DBCS font table		*READ, *ADD
CRTFORMDF	Source file	*USE	*EXECUTE
	Form definition: REPLACE(*NO)		*READ, *ADD
	Form definition: REPLACE(*YES)	Refer to the general rules.	*READ, *ADD
CRTOVL	Source file	*USE	*EXECUTE
	Overlay: REPLACE(*NO)		*READ, *ADD
	Overlay: REPLACE(*YES)	Refer to the general rules.	*READ, *ADD
CRTPAGDFN	Source file	*USE	*EXECUTE
	Page definition: REPLACE(*NO)		*READ, *ADD
	Page definition: REPLACE(*YES)	Refer to the general rules.	*READ, *ADD
CRTPAGSEG	Source file	*USE	*EXECUTE
	Page segment: REPLACE(*NO)		*READ, *ADD
	Page segment: REPLACE(*YES)	Refer to the general rules.	*READ, *ADD
DLTFNTRSC	Font resource	*OBJEXIST	*EXECUTE
DLTFNTTBL	DBCS font table	*CHANGE	*EXECUTE
DLTFORMDF	Form definition	*OBJEXIST	*EXECUTE
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	DBCS font table	*USE	*EXECUTE
RMVFNTTBLE	DBCS 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

Command	Referenced object	Authority needed	
		For object	For library
1 To use individual operations, you must have the authority required by the individual operation.			

Alerts commands

This table lists the specific authorities required for the alerts commands.

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
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

This table lists the specific authorities required for the application development commands.

Command	Referenced object	Authority needed	
		For object	For library
CHGAMTDFT	User profile of user whose AMT defaults are being changed	*OBJMGT, *USE	*EXECUTE
CHGPDMDFT	User profile of user whose PDM defaults are being changed	*OBJMGT, *USE	*EXECUTE
EDTCLU [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

Command	Referenced object	Authority needed	
		For object	For library
FNDSTRAMT	Source part	*READ	*EXECUTE
FNDSTRAMT2	Source part	*READ	*EXECUTE
FNDSTRPDM	Source part	*READ	*EXECUTE
FNDSTRPDM2	Source part	*READ	*EXECUTE
MRGFORMD	Form description	*READ	*EXECUTE
STRAMT [1]			
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]			
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

Command	Referenced object	Authority needed	
		For object	For library
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
WRKLIBAMT[1, 4]			
WRKLIBPDM [1, 4]			
WRKMBRAMT[1]	Source file	*USE	*EXECUTE
WRKMBRPDM [1]	Source file	*USE	*EXECUTE
WRKOBJAMT[1]	File	*READ or Ownership	*EXECUTE
WRKOBJPDM [1]	File	*READ or Ownership	*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
This command requires *ALLOBJ special authority.

Authority collection commands

This table lists the specific authorities required for the authority collection commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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
CHGAUTCOL [1] (Q)			
DLTAUTCOL [1] (Q)			
ENDAUTCOL [1] (Q)			
STRAUTCOL [1] (Q)			

1
You must have *ALLOBJ special authority or be authorized to the Database Security Administrator function of IBM i (QIBM_DB_SECADM) to use this command.

Authority holder commands

This table lists the specific authorities required for the authority holder commands.

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	Refer to the general rules.	Refer to the general rules.

Authorization list commands

This table lists the specific authorities required for the authorization list commands.

Command	Referenced object	Authority needed	
		For object	For QSYS library
ADDAUTLE [1,6]	*AUTL	*AUTLMGT or ownership	*EXECUTE
CHGAUTLE [1,6]	*AUTL	*AUTLMGT or ownership	*EXECUTE
CRTAUTL			
DLTAUTL	*AUTL	Owner or *ALLOBJ	*EXECUTE
DSPAUTL	*AUTL [6]		*EXECUTE
	Output file	Refer to the general rules.	Refer to the general rules.
DSPAUTLDLO	*AUTL	*USE	*EXECUTE
DSPAUTLOBJ	*AUTL [6]	*READ	*EXECUTE
	Output file	Refer to the general rules.	Refer to the general rules.
EDTAUTL [1,6]	*AUTL	*AUTLMGT or ownership	*EXECUTE
RMVAUTLE [1,6]	*AUTL	*AUTLMGT or ownership	*EXECUTE
RTVAUTLE [2,6]	*AUTL	*AUTLMGT or ownership	*EXECUTE
WRKAUTL [3,4,5,6]	*AUTL		

Command	Referenced object	Authority needed	
		For object	For QSYS library

1
You must be the owner or have authorization list management authority.

2
If you 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.

6
If you are authorized to the IBM i Database Security Administrator function (QIBM_DB_SECADM) you do not need the specified authority to the object. However, users authorized to the QIBM_DB_SECADM function cannot add themselves to an authorization list or change their current authority on an authorization list unless they have the authorities required for the operation.

Binding directory commands

This table lists the specific authorities required for the 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

This table lists the specific authorities required for the change request description commands.

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

Command	Referenced object	Authority needed	
		For object	For library
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 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

This table lists the specific authorities required for the 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

This table lists the specific authorities required for the class commands.

Command	Referenced object	Authority needed	
		For object	For library
CHGCLS	Class	*OBJMGT, *OBJOPR	*EXECUTE
CRTCLS	Class		*READ, *ADD

Command	Referenced object	Authority needed	
		For object	For library
DLTCLS	Class	*OBJEXIST	*EXECUTE
DSPCLS	Class	*USE	*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

This table lists the specific authorities required for the 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

This table lists the specific authorities required for the commands related to the operations on command.

Command	Referenced object	Authority needed	
		For object	For library
CHGCMD	Command	*OBJMGT	*EXECUTE
CHGCMDDFT	Command	*OBJMGT, *USE	*EXECUTE
CHGPRXCMD	Proxy command	*OBJMGT	*EXECUTE
CRTCMD	Source file	*USE	*EXECUTE
	Command: REPLACE(*NO)		*READ, *ADD
	Command: REPLACE(*YES)	Refer to the general rules.	Refer to the general rules.
CRTPRXCMD	Proxy command: REPLACE(*NO)		*READ, *ADD
	Proxy command: REPLACE(*YES)	See General Rules on page D-2	See General Rules on page D-2

Command	Referenced object	Authority needed	
		For object	For library
DLTCMD	Command	*OBJEXIST	*EXECUTE
DSPCMD	Command	*USE	*EXECUTE
GENCMDDOC[3]	Command	*USE	*EXECUTE
	Panel group (associated)	*USE	*EXECUTE
	Output file: REPLACE = (*YES)	*ALL	*CHANGE
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.

3
You must have execute (*X) authority to the directories in the path for the generated file, and write and execute (*WX) authorities to the parent directory of the generated file.

Commitment control commands

This table lists the specific authorities required for the 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 associated 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

This table lists the specific authorities required for the 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

This table lists the specific authorities required for the configuration commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, "Commands shipped with public authority *EXCLUDE," on page 355 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
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
	Printer output (QPSRLDSP), if output(*print) is specified	*USE	*EXECUTE
	Output file, if specified	Refer to the general rules.	Refer to the general rules.
	QSYS/QASRRSTO field reference file, if output file is specified and it does not exist	*USE	*EXECUTE
RTVCFGSTS	Object	*OBJOPR	*EXECUTE

Command	Referenced object	Authority needed	
		For object	For library
RTVCFGSRC	Object	*USE	*EXECUTE
	Source file	*OBJOPR, *OBJMGT, *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, 5, 6, 7]	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 object 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 a value other than *NONE for the Allow object differences (ALWOBJDIF) parameter, or RESETSYS(*YES).

6
You must have *IOSYSCFG special authority when the object is a media library and the status is *ALLOCATE or *DEALLOCATE.

7
You must have *IOSYSCFG and *SECADM special authorities to specify GENPTHCERT(*YES).

Configuration list commands

This table lists the specific authorities required for the 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
CRTCFGL [2]	Configuration list		
DLTCFGL	Configuration list	*OBJEXIST	*EXECUTE
DSPCFGL [2]	Configuration list	*USE, *OBJMGT	*EXECUTE

Command	Referenced object	Authority needed	
		For object	For library
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

This table lists the specific authorities required for the connection list commands.

Command	Referenced object	Authority needed	
		For object	For library
DLTCNNL	Connection list	*OBJEXIST	*EXECUTE
DSPCNNL	Connection list	*USE	*EXECUTE
WRKCNNL [1]	Connection list	*OBJOPR	*EXECUTE

1
To use the individual operations, you must have the authority required by the individual operation.

Controller description commands

This table lists the specific authorities required for the 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
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
CHGCTLTAP [2]	Controller description	*CHANGE, *OBJMGT	*EXECUTE

Command	Referenced object	Authority needed	
		For object	For library
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		
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		
Device description (DEV)	*USE	*EXECUTE	
Controller description			
Device description (DEV)	*USE	*EXECUTE	
Connection list (CNNLSTOUT)	*USE	*EXECUTE	
Controller description			
CRTCTLTAP [2]	Device description (DEV)	*USE	*EXECUTE
	Controller description		

Command	Referenced object	Authority needed	
		For object	For library
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 [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 and *IOSYSCFG, or *AUDIT special authority.

Cryptography commands

This table lists the specific authorities required for the cryptography commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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
ADDCKMKSFE	User file	*ADD, *OBJOPR, *READ	
	User library		*EXECUTE
	User directory	*X	
	User stream file	*R	
ADDMSTPART (Q) [1]			
CHKMSTKVV (Q) [1]			
CLRMSTKEY (Q) [1]			
CRTCKMKSF	User library		*ADD, *EXECUTE
DSPCKMKSFE	User file	*OBJOPR, *READ	
	User library		*EXECUTE
GENCKMKSFE	User file	*ADD, *OBJOPR, *READ	
	User library		*EXECUTE

Command	Referenced object	Authority needed	
		For object	For library
RMVCKMKSFE	User file	*DLT, *OBJOPR	
	User library		*EXECUTE
SETMSTKEY (Q) [1]			
TRNCKMKSF	User file	*OBJOPR, *READ, *UPD	
	User library		*EXECUTE

1
You must have *ALLOBJ and *SECADM special authorities to use this command.

Data area commands

This table lists the specific authorities required for the 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	*USE	*EXECUTE
RTVDTAARA [2]	Data area	*USE	*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 even though 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

This table lists the specific authorities required for the 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

This table lists the specific authorities required for the device description commands.

Command	Referenced object	Authority needed	
		For object	For library
CFGDEVASP (Q)[4,8]			
CFGDEVMLB [4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
CHGASPA (Q)			
CHGASPACT (Q)[7]	Device description	*USE	
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
CHGDEVCRP[4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
CHGDEVDSP [4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
	Printer (PRINTER)	*USE	*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

Command	Referenced object	Authority needed	
		For object	For library
CHGDEVNWSH[4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
CHGDEVOPT [4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
CHGDEVPRT [4]	Device description	*CHANGE, *OBJMGT	*EXECUTE
	Validation list (if specified)	*READ	*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
CRTDEVASC [4]	Controller description (CTL)	*USE	*EXECUTE
	Device description		
CRTDEVASP [4]	Device description		*EXECUTE
CRTDEVBSC [4]	Controller description (CTL)	*USE	*EXECUTE
	Device description		
CRTDEVCRP[4]	Device description		*EXECUTE
CRTDEVDSP [4]	Printer description (PRINTER)	*USE	*EXECUTE
	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		
CRTDEVNWSH[4]	Device description		*EXECUTE
CRTDEVOPT [4]	Device description		*EXECUTE
CRTDEVPRT [4]	Controller description (CTL)	*USE	*EXECUTE
	Device description		
	Validation list (if specified)	*READ	*EXECUTE
CRTDEVSNPT [4]	Controller description (CTL)	*USE	*EXECUTE
	Device description		
CRTDEVSNUF [4]	Controller description (CTL)	*USE	*EXECUTE
	Device description		

Command	Referenced object	Authority needed	
		For object	For library
CRTDEVTAP [4]	Controller description (CTL)	*USE	*EXECUTE
	Device description		
DLTDEVD [1]	Device description	*OBJEXIST	*EXECUTE
DSPASPINF	Device description	*USE	
DSPASPSTS	Device description	*USE	
DSPCNNSTS	Device description	*OBJOPR	*EXECUTE
DSPDEVD	Device description	*USE	*EXECUTE
ENDASPBAL (Q)			
ENDDEVRCY	Device description	*USE	*EXECUTE
HLDCMNDEV [2]	Device description	*OBJOPR	*EXECUTE
PRTCMNSEC [4, 5]			
RLSCMNDEV	Device description	*OBJOPR	*EXECUTE
RSMDEVRCY	Device description	*USE	*EXECUTE
SETASPGRP[6]	All device descriptions in ASP group	*USE	
	All the specified libraries in the library list before the library namespace and the library list are changed	*USE	
STRASPBAL (Q)			
TRCASPBAL (Q)			
WRKDEVD [3]	Device description	*OBJOPR	*EXECUTE

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 execute (*EXECUTE) 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. **6** When *CURUSR is specified for the ASP group (ASPGRP) or the Libraries for the current thread (USRLIBL) parameter, you must also have read (*READ) authority to the job description that is listed in your user profile and execute (*EXECUTE) authority to the library where the job description is located. **7** You must have *JOBCTL special authority to run this command. **8** You must have *SERVICE special authority to run this command or must be authorized to the IBM i Service Disk Units function. The Change Function Usage Information (QSYCHFUI) API, with a function ID of QIBM_QYAS_SERVICE_DISKMGMT may also be used to change the list of users who are allowed to work with disk units.			

Device emulation commands

This table lists the specific authorities required for the 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

Command	Referenced object	Authority needed	
		For object	For library
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, workstation device, and workstation 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, printer output, 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

This table lists the specific authorities required for the directory and directory shadowing commands.

These commands do not require any object authorities:			
ADDDIRE [2] ADDDIRSHD [1] CHGSYSDIRA [2] CHGDIRE [3]	CHGDIRSHD [1] CPYFRMDIR [1] CPYTODIR [1] DSPDIRE	ENDDIRSHD [4] RMVDIRE [1] RMVDIRSHD [1] RNMDIRE [2]	STRDIRSHD [4] WRKDIRE [3,5] WRKDIRLOC [1,5] WRKDIRSHD [1,5]

1
You must have *SECADM special authority.

2
You must have *SECADM or *ALLOBJ special authority.

3
A user with *SECADM special authority can work with all directory entries. Users without *SECADM special authority can work only with their own entries.

4
You must have *JOBCTL special authority.

5
To use an individual operation, you must have the authority required by the operation.

Directory server commands

This table lists the specific authorities required for the directory server commands.

Command	Referenced object	Authority needed	
		For object	For library
ADDDIRINST [1]			
CHGDIRSRVA [1]			
CPYTOLDIF [2]	LDIF stream file (if it already exists)	*STMF	*W, *OBJEXIST, *OBJMGT
	Parent directory of LDIF stream file	*DIR	*WX
CPYFRMLDIF [2]	LDIF stream file	*STMF	*R
	Parent directory of LDIF stream file	*DIR	*X
DB2LDIF [2]	LDIF stream file (if it already exists)	*STMF	*W, *OBJEXIST, *OBJMGT
	Parent directory of LDIF stream file	*DIR	*WX
LDIF2DB [2]	LDIF stream file	*STMF	*R
	Parent directory of LDIF stream file	*DIR	*X
RMVDIRINST [1]			

1
You must have *ALLOBJ and *IOSYSCFG special authority.

2
To use this command, you must meet one of the following conditions:

- Have *ALLOBJ and *IOSYSCFG special authorities
- Provide the administrator DN and password
- Be a Directory Server administrator

Disk commands

This table lists the specific authorities required for the disk commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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 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

This table lists the specific authorities required for the display station pass-through commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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
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 signon display, the user profile is the one specified in the remote user (RMTUSER) parameter. For pass-through that uses the normal signon 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 signon procedure, the user profile specified on the signon display on the target system must have authority to this object.

Distribution commands

This table lists the specific authorities required for the distribution commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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)			

Command	Referenced object	Authority needed	
		For object	For library
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

This table lists the specific authorities required for the distribution list commands.

These commands do not require any object authorities:			
ADDDSTLE [1] CHGDSTL[1]	CRTDSTL DLTDSTL [1]	DSPDSTL RMVDSTLE [1]	RNMDSTL [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

This table lists the specific authorities required for the 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[20]	Document list	*ALL [4]	*EXECUTE
DMPDLO [15]			
DSPAUTLDLO	Authorization list	*USE	*EXECUTE
	Document library object	*USE	*EXECUTE
DSPDLOAUD[21]	Output file, if specified	Refer to the general rules.	Refer to the general rules.
DSPDLOAUT	Document library object	*USE or owner	*EXECUTE
DSPDLONAM[22]	Document library object	*USE	*EXECUTE
DSPDOC	Document	*USE	*EXECUTE
DSPFLR	Folder	*USE	*EXECUTE
EDTDLOAUT	Document library object	*ALL or owner	*EXECUTE

Command	Referenced object	Authority needed	
		For object	For library
EDTDOC	Document	*CHANGE	*EXECUTE
FILDOC [2]	Requested file	*USE	*EXECUTE
	Folder	*CHANGE	*EXECUTE
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	Refer to the general rules.	Refer to the general rules.
	To-folder if to-document is new	Refer to the general rules.	Refer to the general rules.
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(*ALL), DLO(*ALL) FLR(*ANY), or DLO(*ALL) FLR(*ANY) MAIL(*YES)[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 (Q)[7, 8, 9]	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	Refer to the general rules.	Refer to the general rules.
	Save file	*USE	*EXECUTE
	Optical file (OPTFILE)[17]	*R	Not applicable
	Path prefix of optical file (OPTFILE)[17]	*X	Not applicable
	Optical volume[19]	*USE	Not applicable
	Tape unit 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[22]	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 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	Refer to the general rules.	Refer to the general rules.
	Optical File (OPTFILE)[17]	*RW	Not applicable
	Parent directory of optical file (OPTFILE)[17]	*WX	Not applicable
	Path Prefix of optical file (OPTFILE)[17]	*X	Not applicable
	Root Directory (/) of volume[17, 18]	*RWX	Not applicable
	Optical Volume[19]	*CHANGE	Not applicable
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	

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** You 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** You must have authority to the object being used as the merge source. For example, if MRGTYPE(*QRY) is specified, you 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 you have at least *USE authority are returned in the document list or output file. **7** You must have *SAVSYS, *ALLOBJ special authority, or have been enrolled in the system distribution directory. **8** You must have *SAVSYS or *ALLOBJ special authority to use the following parameter combination: RSTDLO DLO(*MAIL). **9** You must have *ALLOBJ special authority to specify a value other than *NONE for the Allow object differences (ALWOBJDIF) parameter. **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** You must have *SAVSYS or *ALLOBJ special authority 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.			

Command	Referenced object	Authority needed	
		For object	For library
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. **20** You must have *ALLOBJ special authority when OWNER (*ALL) or OWNER (name) and Name is a different user profile as the caller.			
21 You must have all object (*ALLOBJ) or audit (*AUDIT) special authority to use this command. **22** You must have all object (*ALLOBJ) special authority to use this command when specifying *DST for the object class that is to be located.			

Domain Name System commands

This table lists the specific authorities required for the Domain Name System (DNS) commands.

Command	Referenced object	Authority needed	
		For object	For library
ADDDNSSIG [2]	Existing zone data file	*R	
	Path to existing zone data file	*X	
	Existing keyset directory files	*R	
	Path to existing keyset directory files	*X	
	Existing signed zoned output file	*R	
	Path to existing signed zoned output file	*X	
	Existing entropy source file	*R	
	Path to existing entropy source file	*X	
	Existing output file	*W	
	Path to existing output file	*X	
	Parent of new output file	*RX	
CHKDNSCFG [1]	Existing configuration file	*R	
	Path to existing configuration file	*X	
	Existing output file	*W	
	Path to existing output file	*X	
	Parent of new output file	*RX	

Command	Referenced object	Authority needed	
		For object	For library
CHKDNSZNE [1]	Existing zone file	*R	
	Path to existing zone file	*X	
	Existing output file	*W	
	Path to existing output file	*X	
	Parent of new output file	*RX	
CRTDDNSCFG	Existing entropy source file	*R	
	Path to existing entropy source file	*X	
	Existing output file	*W	
	Path to existing output file	*X	
	Parent of new output file	*RX	
CRTRNDCCFG [1]	Existing entropy source file	*R	
	Path to existing entropy source file	*X	
	Existing output file	*W	
	Path to existing output file	*X	
	Parent of new output file	*RX	
DMPDNSJRNF	Existing output file	*W	
	Path to existing output file	*X	
	Parent of new output file	*RX	
GENDNSDSRR	Existing entropy source file	*R	
	Path to existing entropy source file	*X	
	Existing keyset directory files	*R	
	Path to existing keyset directory files	*X	
	Existing output file	*W	
	Path to existing output file	*X	
	Parent of new output file	*RX	
GENDNSKEY [2]	Existing entropy source file	*R	
	Path to existing entropy source file	*X	
	Existing keyset directory files	*R	
	Path to existing keyset directory files	*X	
	Existing output file	*W	
	Path to existing output file	*X	
	Parent of new output file	*RX	

Command	Referenced object	Authority needed	
		For object	For library
RUNDNSUPD	Existing batch input file	*R	
	Path to existing batch input file	*X	
	Existing key file	*R	
	Path to existing key file	*X	
	Existing output file	*W	
	Path to existing output file	*X	
	Parent of new output file	*RX	
RUNRNDCCMD	Existing RNDC configuration file	*R	
	Path to existing RNDC configuration file	*X	
	Existing key file	*R	
	Path to existing key file	*X	
	Existing output file	*W	
	Path to existing output file	*X	
	Parent of new output file	*RX	
SETDNSRVK [2]	Existing key file	*R	
	Path to existing key file	*X	
	Existing output file	*W	
	Path to existing output file	*X	
	Parent of new output file	*RX	
STRDIGQRY	Existing batch input file	*R	
	Path to existing batch input file	*X	
	Existing trusted key file	*R	
	Path to existing trusted key file	*X	
	Existing key file	*R	
	Path to existing key file	*X	
	Existing output file	*W	
	Path to existing output file	*X	
	Parent of new output file	*RX	
STRHOSTQRY	Existing output file	*W	
	Path to existing output file	*X	
	Parent of new output file	*RX	

1
You must have *IOSYSCFG special authority to run this command.

2
You must have *SECADM special authority to run this command.

Double-byte character set commands

This table lists the specific authorities required for the 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
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

This table lists the specific authorities required for the 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

This table lists the specific authorities required for the 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

This table lists the specific authorities required for the 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
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

This table lists the specific authorities required for the file commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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

Command	Referenced object	Authority needed	
		For object	For library
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
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

Command	Referenced object	Authority needed	
		For object	For library
CHGPFM	Physical file	*OBJMGT or *OBJALTER	*EXECUTE
CHGPFTRG	Physical file	*OBJMGT or *OBJALTER	*EXECUTE
CHGPRTF	Printer output	*OBJOPR, *OBJMGT	*EXECUTE
	Device if device name specified	*OBJOPR	*EXECUTE
CHGSAVF	Save file	*OBJOPR, and (*OBJMGT or *OBJALTER).	*EXECUTE
CHGSRCPF	Source physical file	*OBJMGT or *OBJALTER	*EXECUTE
CHGTAPF	Tape file	*OBJOPR, *OBJMGT	*EXECUTE
	Device if device name specified	*OBJOPR	*EXECUTE
CLRPFM[12]	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)	Refer to the general rules.	Refer to the general rules.
	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)	Refer to the general rules.	Refer to the general rules.
CPYFRMIMPF	From-file	*OBJOPR, *READ	*USE
	To-file (device file)	*OBJOPR, *READ	*USE
	To-file (physical file)	Refer to the general rules.	Refer to the general rules.
	Based-on file if from-file is logical file	*READ	*USE
	command CRTDDMF	*USE	*USE
CPYFRMQRYF [1]	From-file	*OBJOPR, *READ	*EXECUTE
	To-file (device file)	*OBJOPR, *READ	*EXECUTE
	To-file (physical file)	Refer to the general rules.	Refer to the general rules.

Command	Referenced object	Authority needed	
		For object	For library
CPYFRMSTMF	Stream file	*R	
	Directories in stream file path name prefix	*X	
	Target database file, if MBROPT(*ADD) specified	*WX	*X
	Target database file, if MBROPT(*REPLACE or *NONE) specified	*WX, *OBJMGT	*X
	Target database file, if new member created	*WX	*X, *ADD
	Conversion table *TBL used to translate data	*R	*X
	Target save file exists	*RWX, *OBJMGT	*X
	Target save file is created		*RWX
CPYFRMTAP	From-file	*OBJOPR, *READ	*EXECUTE
	To-file (device file)	*OBJOPR, *READ	*EXECUTE
	To-file (physical file)	Refer to the general rules.	Refer to the general rules.
CPYSRCF	From-file	*OBJOPR, *READ	*EXECUTE
	To-file (device file)	*OBJOPR, *READ	*EXECUTE
	To-file (physical file)	Refer to the general rules.	Refer to the general rules.
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)	Refer to the general rules.	Refer to the general rules.
	Based-on file if from-file is logical file	*READ	*USE
	command CRTDDMF	*USE	*USE

Command	Referenced object	Authority needed	
		For object	For library
CPYTOSTMF	Database file or save file	*RX	*X
	Stream file, if it already exists	*W	
	Stream file parent directory, if the stream file does not exist	*WX	
	Stream file path name prefix	*X	
	Database file and stream file, if AUT(*FILE) or AUT(*INDIRFILE) is specified	*OBJMGT	
	Conversion table *TBL used to translate data	*R	*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)	Refer to the general rules.	*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)	Refer to the general rules.	*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)	Refer to the general rules.	*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)	Refer to the general rules.	*READ, *ADD

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
	Printer output: Replace(*NO)		*READ, *ADD, *EXECUTE
	Printer output: Replace(*YES)	Refer to the general rules.	*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)	Refer to the general rules.	*READ, *ADD
	Create Display File (CRTDSPF) command	*OBJOPR	*EXECUTE

Command	Referenced object	Authority needed	
		For object	For library
CRTTAPF	Tape file: REPLACE(*NO)		*READ, *ADD
	Tape file: REPLACE(*YES)	Refer to the general rules.	*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
DSPDBR	Database file	*OBJOPR	*EXECUTE
	Output file, if specified	Refer to the general rules.	Refer to the general rules.
DSPDDMF	DDM file	*OBJOPR	
DSPDTA	Data file	*USE	*EXECUTE
	Program	*USE	*EXECUTE
	Display file	*USE	*EXECUTE
DSPFD [2]	File	*OBJOPR	*EXECUTE
	Output file	Refer to the general rules.	Refer to the general rules.
	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	Refer to the general rules.	Refer to the general rules.
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

Command	Referenced object	Authority needed	
		For object	For library
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 [11]			
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
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	Refer to the general rules.
	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	Not applicable
	Parent Directory of OPTFILE[8]	*WX	Not applicable
	Path Prefix of OPTFILE[8]	*X	Not applicable
	Root Directory (/) of Optical Volume [8,9]	*RWX	Not applicable
	Optical Volume[10]	*CHANGE	Not applicable

Command	Referenced object	Authority needed	
		For object	For library
SAVS36F	From-file	*USE	*EXECUTE
	To-file, when it is a physical file	*ALL	Refer to the general rules.
	Device file or device description	*USE	*EXECUTE
SAVS36LIBM	To-file, when it is a physical file	*ALL	Refer to the general rules.
	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
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
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 before 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.

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.

Command	Referenced object	Authority needed	
		For object	For library
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. **11** You must have *ALLOBJ or *AUDIT special authority to use this command. **12** If the file has active row access control (an active permission) the user must also have *OBJEXIST.			

Filter commands

This table lists the specific authorities required for the 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
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

This table lists the specific authorities required for the finance commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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.

Function usage commands

This table lists the specific authorities required for the function usage commands.

Command	Referenced object	Authority needed	
		For object	For library
CHGFCNUSG[1]			
DSPFCNUSG			
WRKFCNUSG			

1
You must have security administrator (*SECADM) special authority to change the usage of a function.

IBM i graphical operations commands

This table lists the specific authorities required for the IBM i graphical operations commands.

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

Command	Referenced object	Authority needed	
		For object	For library

1 The workstation object is an internal object that is created when you install the IBM i 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 or 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 authorities:

- Own the workstation object.
- Have *ALL authority to the workstation object.
- Have *ALLOBJ special authority.

Graphics symbol set commands

This table lists the specific authorities required for the 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.

High availability commands

This table lists the specific authorities required for the high availability commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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
ADDASPCPYD (Q)[1]	Auxiliary storage pool (ASP) device description	*USE	
	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
ADDCADMRE (Q)[1]	QMRAP1 service program	*USE	

Command	Referenced object	Authority needed	
		For object	For library
ADDCADNODE (Q)[1]	QCSTCRG1 service program	*USE	
	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
ADDCLUMON (Q)[1]	QCSTCTL2 service program	*USE	
ADDCLUNODE (Q)[1]	QCSTCTL service program	*USE	
ADDCRGDEVE (Q)[1]	QCSTCRG1 service program	*USE	
	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
	Exit program	*EXECUTE[2]	*EXECUTE[2]
	User profile to run exit program	*USE	
	Device description	*USE, *OBJMGT	
	Controller description	*USE, *OBJMGT	
	Line description	*USE, *OBJMGT	
	Network server description	*USE, *OBJMGT	
ADDCRGNODE (Q)[1]	QCSTCRG1 service program	*USE	
	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
	Exit program	*EXECUTE[2]	*EXECUTE[2]
	User profile to run exit program	*USE	
	Failover message queue	*OBJOPR, *ADD	*EXECUTE
	Distribute information user queue	*OBJOPR, *ADD	*EXECUTE
ADDDEVDMNE (Q)[1]	QCSTDD service program	*USE	
ADDHACFGD (Q)[1]			
ADDHAPCY (Q)[1]			
ADDHYSSTGD (Q)[1]			
ADDSVCCPYD (Q)[1]	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
	Auxiliary storage pool (ASP) device description	*USE	
	Secure Shell (SSH) key file	*R	
CFGCRGCNR (Q)[1]	QCSTCRG1 service program	*USE	
	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
	Exit program	*EXECUTE[2]	*EXECUTE[2]
	User profile to run exit program	*USE	
	Managed CRG	*CHANGE to each CRG specified on the configuration object list.	

Command	Referenced object	Authority needed	
		For object	For library
CFGGEOMIR (Q)[1,6]			
CHGASPCPYD (Q)[1]	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
	Auxiliary storage pool (ASP) device description	*USE	
CHGASPSSN (Q) [5]	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
	Auxiliary storage pool (ASP) device description	*USE	
CHGCAD (Q)[1]	QCSTCRG1 service program	*USE	
CHGCLU (Q)[1]	QCSTCTL service program	*USE	
CHGCLUMON (Q)[1]	QCSTCTL2 service program	*USE	
CHGCLUNODE (Q)[1]	QCSTCTL service program	*USE	
CHGCLURCY	Cluster resource group	*USE	
		*JOBCTL	
		*SERVICE or Service Trace function	
CHGCLUVER (Q)[1]	QCSTCTL2 service program	*USE	
CHGCRG (Q)[1]	QCSTCRG1 service program	*USE	
	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
	Exit program	*EXECUTE[2]	*EXECUTE[2]
	User profile to run exit program	*USE	
	Device description	*USE, *OBJMGT	
	Failover message queue	*OBJOPR, *ADD	*EXECUTE
	Controller description	*USE, *OBJMGT	
	Line description	*USE, *OBJMGT	
	Network server description	*USE, *OBJMGT	
CHGCRGCNR (Q)[1]	QCSTCRG1 service program	*USE	
	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
	Exit program	*EXECUTE[2]	*EXECUTE[2]
	User profile to run exit program	*USE	
	Managed CRG	For each managed CRG you must have the same authority as the CHGCRG command.	

Command	Referenced object	Authority needed	
		For object	For library
CHGCRGDEVE (Q)[1]	QCSTCRG1 service program	*USE	
	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
	Exit program	*EXECUTE[2]	*EXECUTE[2]
	User profile to run exit program	*USE	
	Device description	*USE, *OBJMGT	
	Controller description	*USE, *OBJMGT	
	Line description	*USE, *OBJMGT	
	Network server description	*USE, *OBJMGT	
CHGCRGPRI (Q)[1]	QCSTCRG2 service program	*USE	
	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
	Exit program	*EXECUTE[2]	*EXECUTE[2]
	User profile to run exit program	*USE	
	Device description	*USE, *OBJMGT	
	Vary configuration (VFYCFG) command	*USE	
	Controller description	*USE, *OBJMGT	
	Line description	*USE, *OBJMGT	
	Network server description	*USE, *OBJMGT	
CHGCSMSSN (Q)[1]	Auxiliary storage pool (ASP) device description	*USE	
	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
CHGHACFGD (Q)[1]			
CHGHAPCY (Q)[1]			
CHGHYSSTGD (Q)[1]			
CHGHYSSTS (Q)[1]			
CHGSVCCPYD (Q)[1]	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
	Auxiliary storage pool (ASP) device description	*USE	
	Secure Shell (SSH) key file	*R	
CHGSVCSSN (Q)[1]	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
	Auxiliary storage pool (ASP) device description	*USE	
	Secure Shell (SSH) key file	*R	

Command	Referenced object	Authority needed	
		For object	For library
CRTCAD (Q)[1]	QCSTCRG1 service program	*USE	
	Cluster resource group		*OBJOPR, *ADD, *READ (QUSRSYS)
CRTCLU (Q)[1]	QCSTCTL service program	*USE	
CRTCRG (Q) [1]	QCSTCRG1 service program	*USE	
	Cluster resource group library		*OBJOPR, *ADD, *READ (QUSRSYS)
	Exit program	*EXECUTE[2]	*EXECUTE[2]
	User profile to run exit program	*USE	
	Device description	*USE, *OBJMGT	
	Distribute information user queue	*OBJOPR, *ADD	*EXECUTE
	Failover message queue	*OBJOPR, *ADD	*EXECUTE
	Controller description	*USE, *OBJMGT	
	Line description	*USE, *OBJMGT	
	Network server description	*USE, *OBJMGT	
CRTCRGCNR (Q)[1]	QCSTCRG1 service program	*USE	
	Cluster resource group library		*OBJOPR, *ADD, *READ (QUSRSYS)
	Exit program	*EXECUTE[2]	*EXECUTE[2]
	User profile to run exit program	*USE	
	Managed CRG	*CHANGE to each CRG specified on the configuration object list.	
DLTCAD (Q)[1]	QCSTCRG1 service program	*USE	
	Cluster resource group	*OBJEXIST, *USE	*EXECUTE (QUSRSYS)
DLTCLU (Q)[1]	QCSTCTL service program	*USE	
DLTCRG[1]	Cluster resource group	*OBJEXIST, *USE	*EXECUTE (QUSRSYS)
DLTCRGCLU (Q)[1]	QCSTCRG1 service program	*USE	
	Cluster resource group	*OBJEXIST, *USE	*EXECUTE (QUSRSYS)
	Exit program	*EXECUTE[2]	*EXECUTE[2]
	User profile to run exit program	*USE	

Command	Referenced object	Authority needed	
		For object	For library
DLTCRGCNR (Q)[1]	QCSTCRG1 service program	*USE	
	Cluster resource group	*OBJEXIST, *USE	*EXECUTE (QUSRSYS)
	Exit program	*EXECUTE[2]	*EXECUTE[2]
	User profile to run exit program	*USE	
DMPCLUTRC	Cluster resource group	*USE	
		*SERVICE or Service Trace function	
DSPASPCPYD (Q)	Auxiliary storage pool (ASP) device description	*USE	
DSPASPSSN (Q)	Auxiliary storage pool (ASP) device description	*USE	
DSPCLUINF			
DSPCRGINF	Cluster resource group	*USE	*EXECUTE (QUSRSYS)
DSPCRGCNR (Q)	Cluster resource group	*USE	*EXECUTE (QUSRSYS)
DSPCSMSSN (Q)[1]	Auxiliary storage pool (ASP) device description	*USE	
DSPHACFGD (Q)			
DSPHAPCY (Q)			
DSPHYSSTGD (Q)[1]			
DSPHYSSTS (Q)[1]			
DSPSVCCPYD (Q)	Auxiliary storage pool (ASP) device description	*USE	
DSPSVCSSN (Q)	Auxiliary storage pool (ASP) device description	*USE	
	Secure Shell (SSH) key file	*R	
ENDASPSSN (Q)[5]	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
	Auxiliary storage pool (ASP) device description	*USE	
ENDCAD (Q)	QCSTCRG2 service program	*USE	
	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
ENDCLUNOD (Q)[1]	QCSTCTL service program	*USE	
ENDCHTSVR (Q)	Authorization list	*CHANGE	

Command	Referenced object	Authority needed	
		For object	For library
ENDCRG (Q)[1]	QCSTCRG2 service program	*USE	
	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
	Exit program	*EXECUTE[2]	*EXECUTE[2]
	User profile to run exit program	*USE	
ENDCRGCNR (Q)[1]	QCSTCRG2 service program	*USE	
	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
	Exit program	*EXECUTE[2]	*EXECUTE[2]
	Managed CRG	For each managed CRG you must have the same authority as the ENDCRG command.	
ENDCSMSSN (Q)[1]	Auxiliary storage pool (ASP) device description	*USE	
	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
ENDSVCSSN (Q)[1]	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
	Auxiliary storage pool (ASP) device description	*USE	
	Secure Shell (SSH) key file	*R	
PRTCADMRE (Q)	QCSTCRG3 service program	*USE	
	QFPADAP1	*USE	
	Cluster Resource Group	*USE	*EXECUTE (QUSRSYS)
RMVASPCPYD (Q)[1]	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
	Auxiliary storage pool (ASP) device description	*USE	
RMVCADMRE (Q)[1]	QMRAP1 service program	*USE	
RMVCADNODE (Q)[1]	QCSTCRG1 service program	*USE	
	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
RMVCLUMON (Q)[1]	QCSTCTL2 service program	*USE	
RMVCLUNODE (Q)[1]	QCSTCTL service program	*USE	

Command	Referenced object	Authority needed	
		For object	For library
RMVCRGDEVE (Q)[1]	QCSTCRG1 service program	*USE	
	Cluster resource group	*CHANGE	*EXECUTE
	Exit program	*EXECUTE[2]	*EXECUTE[2]
	User profile to run exit program	*USE	
	Device description	*USE, *OBJMGT	
	Controller description	*USE, *OBJMGT	
	Line description	*USE, *OBJMGT	
	Network server description	*USE, *OBJMGT	
RMVCRGNODE (Q) [1]	QCSTCRG1 service program	*USE	
	Cluster resource group	*CHANGE, *OBJEXIST	*EXECUTE
	Exit program	*EXECUTE[2]	*EXECUTE[2]
	User profile to run exit program	*USE	
	Device description	*USE, *OBJMGT	
	Controller description	*USE, *OBJMGT	
	Line description	*USE, *OBJMGT	
	Network server description	*USE, *OBJMGT	
RMVDEVDMNE (Q)[1]	QCSTDD service program	*USE	
RMVHACFGD (Q)[1]			
RMVHAPCY (Q)[1]			
RMVHYSSTGD (Q)[1]			
RMVSVCCPYD (Q)[1]	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
	Auxiliary storage pool (ASP) device description	*USE	
	Secure Shell (SSH) key file	*R	
RSTHAPCY (Q)[1]			
RTVASPCPYD	Auxiliary storage pool (ASP) device description	*USE	
RTVASPSSN	Auxiliary storage pool (ASP) device description	*USE	
RTVCLU	QHASM/QHAAPI service program	*USE	
	QCSTCTL1 service program	*USE	

Command	Referenced object	Authority needed	
		For object	For library
RTVCRG	QCSTCTL1 service program	*USE	
	QCSTCRG3 service program	*USE	
	Cluster resource group	*USE	*EXECUTE (QUSRSYS)
RTVCSMSSN (Q)[1]	Auxiliary storage pool (ASP) device description	*USE	
RTVSVCCPYD (Q)	Auxiliary storage pool (ASP) device description	*USE	
RTVSVCSSN (Q)	Auxiliary storage pool (ASP) device description	*USE	
	Secure Shell (SSH) key file	*R	
SAVHAPCY (Q)			
STRASPSSN (Q)	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
	Job description	*READ	*EXECUTE
STRCAD (Q)[1]	QCSTCRG2 service program	*USE	
	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
STRCHTSVR	Authorization list	*CHANGE	
STRCLUNOD (Q)[1]	QCSTCTL service program	*USE	
STRCRG (Q)[1]	QCSTCRG2 service program	*USE	
	Cluster resource group	*CHANGE	*EXECUTE
	Exit program	*EXECUTE[2]	*EXECUTE[2]
	User profile to run exit program	*USE	
	Device description	*USE, *OBJMGT	
	Controller description	*USE, *OBJMGT	
	Line description	*USE, *OBJMGT	
	Network server description	*USE, *OBJMGT	
STRCRGCNR (Q)[1]	QCSTCRG2 service program	*USE	
	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
	Exit program	*EXECUTE[2]	*EXECUTE[2]
	User profile to run exit program	*USE	
	Managed CRG	For each managed CRG you must have the same authority as the STRCRG command.	

Command	Referenced object	Authority needed	
		For object	For library
STRCSMSSN (Q)[1]	Auxiliary storage pool (ASP) device description	*USE	
	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
STRSVCSSN (Q)[1]	Cluster resource group	*CHANGE	*EXECUTE (QUSRSYS)
	Auxiliary storage pool (ASP) device description	*USE	
	Secure Shell (SSH) key file	*R	
WRKASPCPYD (Q)	Auxiliary storage pool (ASP) device description	*USE	
WRKCADMRE (Q)	Cluster resource group	*CHANGE	*EXECUTE
	QCLUSTER user profile	*USE	
WRKCLU[4]	Cluster resource group	*USE	*EXECUTE
WRKHACFGD (Q)			
WRKHAPCY (Q)			
WRKHYSSTS (Q)[1]			

1
You must have *IOSYSCFG special authority to use this command.

2
The authority applies to calling user profile and user profile to run exit program.

3
The calling user profile is granted *CHANGE and *OBJEXIST authority to the cluster resource group.

4
You must have *SERVICE special authority or be authorized to the IBM i Service Trace Function through Application Administration in IBM Navigator for i. The Change Function Usage (CHGFCNUSG) command, 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.

5
You must have *JOBCTL special authority to use this command.

6
You must have *SERVICE special authority or be authorized to the IBM i Service Disk Units Function through Application Administration in IBM Navigator for i. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_QYAS_SERVICE_DISKMGMT, can also be used to change the list of users that are allowed to work with disk units.

Host server commands

This table lists the specific authorities required for the host server commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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)	

Image catalog commands

This table lists the specific authorities required for the image catalog commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, "Commands shipped with public authority *EXCLUDE," on page 355 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	Authority needed	
			For object	For library[1]
ADDIMGCLGE	Image catalog	*IMGCLG	*CHANGE	*EXECUTE
	Image catalog directory path prefix	*DIR	*X	
	Device name when FROMDEV specified	*DEVD	*USE	
	Image file when FROMFILE specified	*STMF	*R, *OBJMGT	
	Image file path prefix when FROMFILE specified	*DIR	*X	
	Image file parent directory when FROMFILE specified	*DIR	*RX	
CHGIMGCLG	Image catalog	*IMGCLG	*CHANGE	*EXECUTE
	Image catalog directory path prefix	*DIR	Refer to the general rules	
	New image catalog directory path prefix when DIR parameter specified	*DIR	Refer to the general rules	
CHGIMGCLGE	Image catalog	*IMGCLG	*CHANGE	*EXECUTE
	Image catalog directory path prefix	*DIR	Refer to the general rules	
CRTIMGCLG	QUSRSYS	*LIB		*READ, *ADD
	Image catalog if DIR(*REFIMGCLG) specified	*IMGCLG	*USE	*OBJOPR, *READ, *ADD, *EXECUTE
	Image catalog directory path prefix[2]	*DIR	Refer to the general rules	
DLTIMGCLG	Image catalog	*IMGCLG	*OBJEXIST	*EXECUTE
	Image catalog directory path prefix	*DIR	Refer to the general rules	
LODIMGCLG	Image catalog	*IMGCLG	*USE	*EXECUTE
	Image catalog when WRTPTC(*ALL) or WRTPTC(*NONE) is specified	*IMGCLG	*CHANGE	*EXECUTE
	Virtual device	*DEVD	*USE	
	Image catalog directory path prefix	*DIR	Refer to the general rules	
LODIMGCLGE	Image catalog	*IMGCLG	*USE	*EXECUTE
	Image catalog directory path prefix	*DIR	Refer to the general rules	
RMVIMGCLGE	Image catalog	*IMGCLG	*CHANGE	*EXECUTE
	Image catalog directory path prefix	*DIR	Refer to the general rules	

Command	Referenced object	Object type	Authority needed	
			For object	For library[1]
RTVIMGCLG	Image catalog	*IMGCLG	*USE	*EXECUTE
	Device description if DEV parameter specified	*DEVD	*USE	
STRNETINS (Q)	Network optical device	*DEVD	*USE	
VFYIMGCLG	Image catalog	*IMGCLG	*USE	*EXECUTE
	Virtual device	*DEVD	*USE	
	Image catalog directory path prefix	*DIR	Refer to the general rules	
WRKIMGCLG	Image catalog	*IMGCLG	*USE	*EXECUTE
WRKIMGCLGE	Image catalog	*IMGCLG	*USE	*EXECUTE

1
The library that image catalog objects reside in is QUSRSYS.

2
If a directory is created, you also need write (*W) authority to the directory to contain the new directory.

Integrated file system commands

This table lists the specific authorities required for the integrated file system commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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 when LNKTYPE(*HARD) is specified	*STMF	QOpenSys, "root" (/),UDFS	*OBJEXIST
	Parent of new link	*DIR	QOpenSys, "root" (/), UDFS	*WX
	Path prefix	Refer to the general rules.		

Command	Referenced object	Object type	File system	Authority needed for object[1]
CHGATR	Object when setting an attribute other than *USECOUNT, *ALWCKPWRT, *DISKSTGOPT, *MAINSTGOPT, *ALWSAV, *SCAN, *CRTOBJSCAN, *SETUID, *SETGID, *RSTRDRNMUNL, *CRTOBJAUD, *INHCKPWRT	Any	All except QSYS.LIB	*W
	Object when setting *USECOUNT, *DISKSTGOPT, *MAINSTGOPT, *ALWSAV, *INHCKPWRT	Any	All except QSYS.LIB	*OBJMGT
		*FILE	QSYS.LIB	*OBJOPR, *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
	Object when setting the following attributes: *CRTOBJSCAN or *SCAN[26]	*DIR and *STMF	QOpenSys, "root" (/), UDFS	
	Object when setting the following attributes: *SETUID, *SETGID, *RSTDRNMUNL	Any	All except QSYS.LIB and QDLS	Ownership [15]
	*CRTOBJAUD[9]			
	Path prefix [9]	Refer to the general rules.		
CHGAUD [4]				
CHGAUT	Object	All	QOpenSys, "root" (/), UDFS	Ownership[15]
			QSYS.LIB, QOPT[11]	Ownership or *ALLOBJ
			QDLS	Ownership, *ALL, or *ALLOBJ
	Optical volume	*DDIR	QOPT[8]	*CHANGE
	Directory that contains objects when SUBTREE(*ALL) is specified	Any directory or library	All	*RX
CHGCURDIR	Object	Any directory		*R
	Optical volume	*DDIR	QOPT[8]	*X
	Path prefix	Refer to the general rules.		

Command	Referenced object	Object type	File system	Authority needed for object[1]
CHGOWN [24]	Object	All	QSYS.LIB	*OBJEXIST
		*FILE, *LIB, *SBSD	QSYS.LIB	*OBJEXIST, *OBJOPR
		All	QOpenSys, "root" (/), UDFS	Ownership and *OBJEXIST[15]
		All	QDLS	Ownership or *ALLOBJ
			QOPT[11]	Ownership or *ALLOBJ
CHGOWN[24]	User profile of old owner—all except QOPT, QDLS	*USRPRF	All	*DLT
	User profile of new owner—all except QOPT	*USRPRF	All	*ADD
	Optical volume	*DDIR	QOPT[8]	*CHANGE
	Directory that contains objects when SUBTREE(*ALL) is specified	Any directory or library	All	*RX
CHGPGP	Object	All	QSYS.LIB	*OBJEXIST
		*FILE, *LIB, *SBSD	QSYS.LIB	*OBJEXIST, *OBJOPR
		All	QOpenSys, "root" (/), UDFS	Ownership[5, 15]
		All	QDLS	Ownership or *ALLOBJ
			QOPT[11]	Ownership or *ALLOBJ
CHGPGP	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
	Directory that contains objects when SUBTREE(*ALL) is specified	Any directory or library	All	*RX

Command	Referenced object	Object type	File system	Authority needed for object[1]
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 *ALLOBJ or Ownership
	Path, if not the user who checked out	*DIR	QOpenSys, "root" (/), UDFS	*X
	Directory that contains objects when SUBTREE(*ALL) is specified	Any directory	All	*RX
	Path prefix	Refer to the general rules.		
CHKOUT	Object	*STMF	QOpenSys, "root" (/), UDFS	*W
		*DOC	QDLS	*W
	Directory that contains objects when SUBTREE(*ALL) is specified	Any directory	All	*RX
	Path prefix	Refer to the general rules.		

Command	Referenced object	Object type	File system	Authority needed for object[1]
CPY[25]	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
		*LIB	QSYS.LIB	*RW, *OBJMGT, *OBJEXIST
		*FILE (PF or LF)	QSYS.LIB	*RW, *OBJMGT, *OBJEXIST
		*DOC	QDLS	*RWX, *ALL
	Directory being copied that contains objects when SUBTREE(*ALL) is specified, so that its contents are copied	*DIR	QOpenSys, "root" (/), UDFS	*RX, *OBJMGT
CPY[25]	Path (target), parent directory of destination object	*FILE	QSYS.LIB	*RX, *OBJMGT
		*LIB	QSYS.LIB	*RX, *ADD
		*DIR	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

Command	Referenced object	Object type	File system	Authority needed for object[1]
CPY[25]	Parent directory of origin object	*DIR	QOpenSys, "root" (/), UDFS	*X
		*FLR	QDLS	*X
		Others	QSYS.LIB	*RX
		*DDIR	QOPT[11]	*X
	Path prefix (target destination)	*LIB	QSYS.LIB	*WX
		*DIR	QOpenSys, "root" (/), UDFS	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
	Path prefix (origin object)	*DDIR	QOPT[11]	*X
CPYFRMSTMF	See "File commands" on page 417			
CPYTOSTMF	See "File commands" on page 417			
CRTDIR[21, 22]	Parent directory	*DIR	QOpenSys, "root" (/), UDFS	*WX
		*FLR	QDLS	*CHANGE
		*FILE	QSYS.LIB	*RX, *ADD
		Any		*ADD
		*DDIR	QOPT[11]	*WX
CRTDIR	Path prefix	Refer to the general rules.		
	Optical volume	*DDIR	QOPT[8]	*CHANGE
CVTDIR (Q)[16]				
DSPATR	Directories in path prefix	Any directory	All	*X
	Directory when pattern is specified (* or ?)	Any directory	All	*RX
	Directory that contains objects when SUBTREE(*ALL) is specified	Any directory	All	*RX
	Object when extended attributes are present	Any	All	*R
DSPAUT	Object	All	QDLS	*ALL
		All	All others	*OBJMGT or ownership
		ALL	QOPT[11]	None
	Optical volume	*DDIR	QOPT[8]	*USE
	Path prefix	Refer to the general rules.		

Command	Referenced object	Object type	File system	Authority needed for object[1]
DSPCURDIR	Path prefix	*DIR	QOpenSys, "root" (/), UDFS	*RX
		*FLR	QDLS	*RX
		*LIB, *FILE	QSYS.LIB	*RX
		*DIR		*R
		*DDIR	QOPT[11]	*RX
DSPCURDIR	Current directory	*DIR	QOpenSys, "root" (/), UDFS	*X
		*LIB, *FILE	QSYS.LIB	*X
		*FLR	QDLS	*X
		*DIR		*R
		*DDIR	QOPT[11]	*X
	Optical volume	*DDIR*	QOPT[8]	*USE
DSPF	Database file	*FILE	QSYS.LIB	*USE
	Database file library	*LIB	QSYS.LIB	*EXECUTE
	Stream file	*STMF	QOpenSys, "root" (/), UDFS	*R
		*USRSPC	QSYS.LIB	*USE
	Path prefix	Refer to the general rules.		
DSPLNK	Any	Any	"root" (/), QOpenSys, UDFS QSYS.LIB[27], QDLS, QOPT[11]	None
	File, Option 12 (Work with Links)	*STMF, *SYMLNK, *DIR, *BLKSF, *SOCKET	"root" (/), QOpenSys, UDFS	*R

Command	Referenced object	Object type	File system	Authority needed for object[1]
DSPLNK	Symbolic link object	*SYMLNK	"root" (/), QOpenSys, UDFS	None
	Optical volume	*DDIR	QOPT[8]	*USE
	Parent directory of referenced object - No Pattern [13]	*DIR	"root" (/), QOpenSys, UDFS	*X
		*LIB, *FILE	QSYS.LIB[27]	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R
DSPLNK	Parent directory of referenced object - Pattern specified[13]	*DIR	"root" (/), QOpenSys, UDFS	*R
		*LIB, *FILE	QSYS.LIB[27]	*R
		*FLR	QDLS	*R
		*DDIR	QOPT[11]	*R
		*DDIR		*R
	Parent directory of referenced object- Option 8 (Display Attributes)	*DIR	"root" (/), QOpenSys, UDFS	*X
		*LIB, *FILE	QSYS.LIB[27]	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R
DSPLNK	Parent directory of referenced object - Option 12 (Work with Links)	*DIR	"root" (/), QOpenSys, UDFS	*RX
		*SYMLNK	"root" (/), QOpenSys, UDFS	*X
		*LIB, *FILE	QSYS.LIB[27]	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R

Command	Referenced object	Object type	File system	Authority needed for object[1]
DSPLNK	Prefix of parent referenced object - No Pattern [13]	*DIR	"root" (/), QOpenSys, UDFS	*X
		*LIB *FILE	QSYS.LIB[27]	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R
DSPLNK	Prefix of parent referenced object - Pattern specified[13]	*DIR	"root" (/), QOpenSys, UDFS	*X
		*LIB *FILE	QSYS.LIB[27]	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R
DSPLNK	Prefix of parent referenced object - Option 8 (Display Attributes)	*DIR	"root" (/), QOpenSys, UDFS	*RX
		*LIB, *FILE	QSYS.LIB[27]	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R
DSPLNK	Prefix of parent referenced object - Option 12 (Work with Links)	*DIR	"root" (/), QOpenSys, UDFS	*RX
		*SYMLNK	"root" (/), QOpenSys, UDFS	*X
		*LIB, *FILE	QSYS.LIB[27]	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R

Command	Referenced object	Object type	File system	Authority needed for object[1]
DSPLNK	Relative Path Name [14] : Current working directory containing object -No Pattern[13]	*DIR	"root" (/), QOpenSys, UDFS	*RX
		*LIB, *FILE	QSYS.LIB[27]	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*RX
		*DDIR		*R
	Relative Path Name [14] : Current working directory containing object -Pattern Specified[13]	*DIR	"root" (/), QOpenSys, UDFS	*RX
		*LIB, *FILE	QSYS.LIB[27]	*RX
		*FLR	QDLS	*RX
		*DDIR	QOPT[11]	*RX
		*DDIR		*R
DSPLNK	Relative Path Name [14] : Prefix of current working directory containing object -No Pattern [13]	*DIR	"root" (/), QOpenSys, UDFS	*RX
		*LIB, *FILE	QSYS.LIB[27]	*RX
		*FLR	QDLS	*RX
		*DDIR	QOPT[11]	*RX
		*DDIR		*R
DSPLNK	Relative Path Name [14] : Prefix of current working directory containing object -Pattern specified[13]	*DIR	"root" (/), QOpenSys, UDFS	*RX
		*LIB *FILE	QSYS.LIB[27]	*RX
		*FLR	QDLS	*RX
		*DDIR	QOPT[11]	*RX
		*DDIR		*R
DSPMFSINF	Object	Any	Any	None
	Path Prefix	Refer to the general rules.		

Command	Referenced object	Object type	File system	Authority needed for object[1]
EDTF	Database file, existing member	*FILE	QSYS.LIB	*CHANGE
	Database file library	*LIB	QSYS.LIB	*EXECUTE
	Database file, new member	*FILE	QSYS.LIB	*CHANGE, *OBJMGT
	Database file library, new member	*LIB	QSYS.LIB	*EXECUTE, *ADD
	Stream file, existing file	*STMF	QOpenSys, "root" (/), UDFS	*R
	User space	*USRSPC	QSYS.LIB	*CHANGE
	Parent directory when creating a new stream file	*DIR	QOpenSys, "root" (/), UDFS	*WX
	Path prefix	Refer to the general rules.		
ENDJRN	Object	*DIR if Subtree (*ALL)	QOpenSys, "root" (/), UDFS	*R, *X, *OBJMGT
		*DIR if Subtree (*NONE), *SYMLNK, *STMF	QOpenSys, "root" (/), UDFS	*R, *OBJMGT
		*DTAARA, *DTAQ	QSYS.LIB	*OBJOPR, *READ, *OBJMGT
	Parent Directory	*DIR	QOpenSys, "root" (/), UDFS	*X
		*LIB	QSYS.LIB	*X
	Journal	*JRN	QSYS.LIB	*OBJMGT, *OBJOPR
	Path Prefix	Refer to the general rules.		

Command	Referenced object	Object type	File system	Authority needed for object[1]
MOV[19]	Object moved within same file system	*DIR	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
MOV	Path (source), parent directory	*DIR	QOpenSys, "root" (/), UDFS	*WX
		*FLR	QDLS	*RWX
		*FILE	QSYS.LIB, "root" (/)	*RX, *OBJEXIST
		others	QOpenSys, "root" (/)	*RWX
	Path (target), parent directory	*DIR	QSYS.LIB	*WX
		*FLR	QDLS	*CHANGE (*RWX)
		*FILE	QSYS.LIB	*X, *ADD, *DLT, *OBJMGT
		*LIB	QSYS.LIB	*RWX
		*DDIR	QOPT[11]	*WX
MOV	Path prefix (target)	*LIB	QSYS.LIB	*X, *ADD
		*FLR	QDLS	*X
		*DIR	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	Not applicable
		*DSTMF	QOPT[11]	*RW

Command	Referenced object	Object type	File system	Authority needed for object[1]
MOV	Moved into QSYS *MBR	*STMF	QOpenSys, "root" (/), UDFS	*R, *OBJMGT, *OBJEXIST
		*DOC	QDLS	*ALL
		*DSTMF	QOPT[11]	*RW
MOV	Optical volume (Source and Target)	*DDIR	QOPT[8]	*CHANGE
	Path (source) moved across file systems, parent directory	*DIR	QOpenSys, "root" (/), UDFS	*WX
		*FLR	QDLS	*X
		*FILE	QSYS. LIB	ownership, *RX, *OBJEXIST
		*DDIR	QOPT[11]	*WX
	Path Prefix	Refer to the general rules.		
RCLLNK[16]				
RLSIFSLCK [18]	object	*STMF	"root" (/), QOpenSys, UDFS	*R
	Path prefix	Refer to the general rules.		
RMVDIR[19,20]	Directory	*DIR	QOpenSys, "root" (/), UDFS	*OBJEXIST
		*LIB	QSYS.LIB	*RX, *OBJEXIST
		*FILE	QSYS.LIB	*OBJOPR, *OBJEXIST
		*FLR	QDLS	*ALL
		*DDIR	QOPT[11]	*W
RMVDIR	Parent directory	*DIR	QOpenSys, "root" (/), UDFS	*WX
		*FLR	QDLS	*X
		*LIB, *FILE	QSYS.LIB	*X
		*DDIR	QOPT[11]	*WX
	Directory that contains objects when SUBTREE(*ALL) is specified	Any directory	All	*RX
	Optical volume	*DDIR	QOPT[8]	*CHANGE
	Path Prefix	Refer to the general rules.		

Command	Referenced object	Object type	File system	Authority needed for object[1]
RMVLNK[19]	Object	*DOC	QDLS	*ALL
		*MBR	QSYS.LIB	
		*FILE	QSYS.LIB	*OBJOPR, *OBJEXIST
		*JRNRCV	QSYS.LIB	*OBJEXIST, *R
		other	QSYS.LIB	*OBJEXIST
		*DSTMF	QOPT[11]	*W
		Any	QOpenSys, "root" (/), UDFS	*OBJEXIST
RMVLNK	Parent Directory	*FLR	QDLS	*X
		*FILE	QSYS.LIB	*X, *OBJEXIST
		*LIB	QSYS.LIB	*X
		*DIR	QOpenSys, "root" (/), UDFS	*WX
		*DDIR	QOPT[11]	*WX
	Optical volume	*DDIR	QOPT[8]	*CHANGE
	Path prefix	Refer to the general rules.		
RNM[19]	Object	*DIR	QOpenSys, "root" (/), UDFS	*OBJMGT, *W
		Not *DIR	QOpenSys, "root" (/), UDFS	*OBJMGT
		*DOC, *FLR	QDLS	*ALL
		*MBR	QSYS.LIB	Not applicable
		*FILE	QSYS.LIB	*OBJMGT, *OBJOPR
		others	QSYS.LIB	*OBJMGT
		*DSTMF	QOPT[11]	*W
	Optical Volume (Source and Target)	*DDIR	QOPT[8]	*CHANGE

Command	Referenced object	Object type	File system	Authority needed for object[1]
RNM	Parent directory	*DIR	QOpenSys, "root" (/), UDFS	*WX
		*FLR	QDLS	*CHANGE (*RWX)
		*FILE	QSYS.LIB	*X, *OBJMGT
		*LIB	QSYS.LIB	*X, *UPD
		*DDIR	QOPT[11]	*WX
	Path prefix	*LIB	QSYS.LIB	*X, *UPD
		Any	QOpenSys, "root" (/), UDFS, QDLS	*X
RST (Q) [23, 28, 30]	Object, if it exists[2]	Any	QOpenSys, "root" (/), UDFS	*W, *OBJEXIST
			QSYS.LIB	Varies [10]
			QDLS	*ALL
	Path prefix	Refer to the general rules.		
	Parent directory created by the restore operation due to CRTPRNDIR(*YES)[2]	*DIR	QOpenSys, "root" (/), UDFS	*WX
	Parent directory owner specified on parameter PRNDIROWN[2, 6]	*USRPRF	QSYS.LIB	*ADD
RST (Q)	Parent directory of object being restored[2]	*DIR	QOpenSys, "root" (/), UDFS	*WX
	Parent directory of object being restored, if the object does not exist[2]	*FLR	QDLS	*CHANGE
		*DIR		*OBJMGT, *OBJALTER, *READ, *ADD, *UPD
	User profile owning new object being restored[2]	*USRPRF	QSYS.LIB	*ADD
	Tape unit, optical unit, or save file	*DEVD, *FILE	QSYS.LIB	*RX
	Media definition	*MEDDFN	QSYS.LIB	*USE

Command	Referenced object	Object type	File system	Authority needed for object[1]
RST (Q)	Library for device description, media definition, or save file	*LIB	QSYS.LIB	*EXECUTE
	Output file, if specified	*STMF	QOpenSys, "root" (/), UDFS	*W
		*USRSPC	QSYS.LIB	*RWX
	Path prefix of output file	*DIR	QOpenSys, "root" (/), UDFS	*X
		*LIB	QSYS.LIB	*RX
RST (Q)	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	QOpenSys, "root" (/), UDFS, QDLS, QOPT[11]	*RX
		*DDIR	QOPT[11]	*RX
		*FLR	QDLS	*RX
		*LIB, *FILE	QSYS.LIB	*RX
		Any		*R
RTVCURDIR	Current directory	*DIR	QOpenSys, "root" (/), UDFS, QOPT [11]	*X
		*DDIR	QOPT[11]	*X
		*LIB, *FILE	QSYS.LIB	*X
		*FLR	QDLS	*X
		Any		*R
SAV[29]	Object[2]	Any	QOpenSys, "root" (/), UDFS	*R, *OBJEXIST
			QSYS.LIB	Varies [10]
			QDLS	*ALL
	Path prefix	Refer to the general rules.		
	Tape unit, optical unit	*DEVD	QSYS.LIB	*RX
	Media definition	*MEDDFN	QSYS.LIB	*USE

Command	Referenced object	Object type	File system	Authority needed for object[1]
SAV	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, media definition, save file, or save-while-active message queue	*LIB	QSYS.LIB	*EXECUTE
SAV	Output file, if specified	*STMF	QOpenSys, "root" (/), UDFS	*W
		*USRSPC	QSYS.LIB	*RWX
	Path prefix of output file	*DIR	QOpenSys, "root" (/), UDFS	*X
		*LIB	QSYS.LIB	*RX
SAV	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.			
STATFS	Object	Any	Any	None
	Path Prefix	Refer to the general rules.		

Command	Referenced object	Object type	File system	Authority needed for object[1]
STRJRN	Object	*DIR if Subtree (*ALL)	QOpenSys, "root" (/), UDFS	*R, *X, *OBJMGT
		*DIR if subtree (*NONE), *SYMLNK, *STMF	QOpenSys, "root" (/), UDFS	*R, *OBJMGT
		*DTAARA, *DTAQ	QSYS.LIB	*OBJOPR, *READ, *OBJMGT
	Parent Directory	*DIR	QOpenSys, "root" (/), UDFS	*X
		*LIB	QSYS.LIB	*X
	Journal	*JRN	QSYS.LIB	*OBJMGT, *OBJOPR
	Path Prefix	Refer to the general rules.		
WRKAUT[6, 7]	Object	*DOC or *FLR	QDLS	*ALL
		All	not QDLS	*OBJMGT or ownership
		*DDIR and *DSTMF	QOPT[11]	*NONE
	Optical volume	*DDIR	QOPT[8]	*USE
	Path prefix	Refer to the general rules.		
WRKLNK	Any	Any	"root" (/), QOpenSys, UDFS, QSYS.LIB[27], QDLS, QOPT[11]	None
	File, Option 12 (Work with Links)	*STMF, *SYMLNK, *DIR, *BLKSF, *SOCKET	"root" (/), QOpenSys, UDFS	*R
	Symbolic link object	*SYMLNK	"root" (/), QOpenSys, UDFS	None
	Optical volume	*DDIR	QOPT[8]	*USE

Command	Referenced object	Object type	File system	Authority needed for object[1]
WRKLNK	Parent directory of referenced object - No Pattern [13]	*DIR	"root" (/), QOpenSys, UDFS	*X
		*LIB, *FILE	QSYS.LIB[27]	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R
WRKLNK	Parent directory of referenced object - Pattern Specified	*DIR	"root" (/), QOpenSys, UDFS	*R
		*LIB *FILE	QSYS.LIB[27]	*R
		*FLR	QDLS	*R
		*DDIR	QOPT[11]	*R
		*DDIR		*R
WRKLNK	Parent directory of referenced object- Option 8 (Display Attributes)	*DIR	"root" (/), QOpenSys, UDFS	*X
		*LIB *FILE	QSYS.LIB[27]	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R
WRKLNK	Parent directory of referenced object - Option 12 (Work with Links)	*DIR	"root" (/), QOpenSys, UDFS	*RX
		*SYMLNK	"root" (/), QOpenSys, UDFS	*X
		*LIB *FILE	QSYS.LIB[27]	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R
WRKLNK	Prefix of parent referenced object - No Pattern [13]	*DIR	"root" (/), QOpenSys, UDFS	*X
		*LIB *FILE	QSYS.LIB[27]	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R

Command	Referenced object	Object type	File system	Authority needed for object[1]
WRKLNK	Prefix of parent referenced object - Pattern specified[13]	*DIR	"root" (/), QOpenSys, UDFS	*X
		*LIB, *FILE	QSYS.LIB[27]	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R
WRKLNK	Prefix of parent referenced object - Option 8 (Display Attributes)	*DIR	"root" (/), QOpenSys, UDFS	*RX
		*LIB, *FILE	QSYS.LIB[27]	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R
WRKLNK	Prefix of parent referenced object - Option 12 (Work with Links)	*DIR	"root" (/), QOpenSys, UDFS	*RX
		*SYMLNK	"root" (/), QOpenSys, UDFS	*X
		*LIB, *FILE	QSYS.LIB[27]	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*X
		*DDIR		*R
WRKLNK	Relative Path Name [14]: Current working directory containing object -No Pattern[13]	*DIR	"root" (/), QOpenSys, UDFS	*RX
		*LIB *FILE	QSYS.LIB[27]	*X
		*FLR	QDLS	*X
		*DDIR	QOPT[11]	*RX
		*DDIR		*R
	Relative Path Name [14] : Current working directory containing object -Pattern Specified[13]	*DIR	"root" (/), QOpenSys, UDFS	*RX
		*LIB *FILE	QSYS.LIB[27]	*RX
		*FLR	QDLS	*RX
		*DDIR	QOPT[11]	*RX
		*DDIR		*R

Command	Referenced object	Object type	File system	Authority needed for object[1]
WRKLNK	Relative Path Name [14] : Prefix of current working directory containing object -No Pattern [13]	*DIR	"root" (/), QOpenSys, UDFS	*RX
		*LIB, *FILE	QSYS.LIB[27]	*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	"root" (/), QOpenSys, UDFS	*RX
		*LIB, *FILE	QSYS.LIB[27]	*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 . 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 If the profile that is specified using the PRNDIROWN parameter is not the user doing the restore operation, *SAVSYS or *ALLOBJ special authority is required.

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.

9 The user must have *AUDIT special authority to change the *CRTOBJAUD attribute, and the user does not need any of the normal path name prefix authorities (*X and *R).

10 Authority required varies by the command used. See the respective SAVOBJ or RSTOBJ command for the required authority.

Command	Referenced object	Object type	File system	Authority needed for object[1]
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 asterisk (*) 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 *ALLOBJ special authority to use this command. **17** In the above table, QSYS.LIB refers to independent ASP QSYS.LIB file systems as well as QSYS.LIB file system. **18** To use this command, you must have *IOSYSCFG special authority. **19** If the restricted renames and unlinks attribute (also known as S_ISVTX bit) is on for a directory, it will restrict unlinking objects from that directory unless one of these authorities is met: • The user has all object (*ALLOBJ) special authority. • The user is the owner of the object being unlinked. • The user is the owner of the directory. **20** If RMVLNK (*YES) is specified, the user must also have *OBJEXIST authority to all objects in the specified directory.				

Command	Referenced object	Object type	File system	Authority needed for object[1]
21 For QSYS.LIB, "root" (/), QOpenSys, and user-defined file systems, the audit (*AUDIT) special authority is required if a value other than *SYSVAL is specified for the CRTOBJAUD parameter. **22** The user must have all object (*ALLOBJ) and security administrator (*SECADM) special authorities to specify a value for the Scanning option for objects (CRTOBJSCAN) parameter other than *PARENT. **23** You must have *ALLOBJ special authority to specify a value other than *NONE for the Allow object differences (ALWOBJDIF) parameter. Also, you must have *SAVSYS or *ALLOBJ special authority to specify *UDFS as the value for the RBDMFS parameter. **24** The user must have all object (*ALLOBJ) and security administrator (*SECADM) special authority when changing the owner of a stream file (*STMF) with an attached Java program whose authority checking while the program is running includes the user and the owner. **25** The user must have all object (*ALLOBJ) and security administrator (*SECADM) special authority when copying a stream file (*STMF) with an attached Java program whose authority checking includes the user and the owner.				
26 The user must have all object (*ALLOBJ) and security administrator (*SECADM) special authority to specify the *CRTOBJSCAN and *SCAN attributes. **27** When you display the contents of the /QSYS.LIB directory, user profile (*USRPRF) objects to which the caller does not have any authority (such as *EXCLUDE) are not returned. **28** The user must have *ALLOBJ special authority to specify *YES for the PVTAUT parameter. **29** The user must have *ALLOBJ or *SAVSYS special authority to specify *YES for the PVTAUT parameter. **30** You must have *SAVSYS or *ALLOBJ special authority to specify *UDFS as the value for the RBDMFS parameter.				

Interactive data definition commands

This table lists the specific authorities required for the 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

Command	Referenced object	Authority needed	
		For object	For library
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

This table lists the specific authorities required for the Internetwork Packet Exchange (IPX) commands.

Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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
DLTIPXD	IPX description	*OBJEXIST	*EXECUTE
DSPIPXD	IPX description	*USE	*EXECUTE
WRKIPXD	IPX description	*OBJOPR	*EXECUTE

Information search index commands

This table lists the specific authorities required for the information search index commands.

Command	Referenced object	Authority needed	
		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

Command	Referenced object	Authority needed	
		For object	For library
WRKSCHIDX [1]	Search index	*ANY	*USE
WRKSCHIDXE	Search index	*USE	*USE

IPL attribute commands

This table lists the specific authorities required for the IPL attribute commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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.

Java commands

This table lists the specific authorities required for the Java commands.

Command	Referenced object	Authority needed	
		For object	For library
ANZJVM	QSYS/STRSRVJOB command	*USE	
	QSYS/STRDBG command	*USE	
DSPJVMJOB[1]	Java Virtual Machine jobs		
GENJVMDMP [1]			
PRTJVMJOB [1]			
WRKJVMJOB [1]			
1 You must have *JOBCTL special authority to use this command.			

Job commands

This table lists the specific authorities required for the Job commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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
	Libraries in the library list (system, current, and user)[7]	*USE	
	User profile in job description [10]	*USE	
	Sort sequence table [7]	*USE	*EXECUTE
	Message queue [10]	*USE, *ADD	*EXECUTE
	Job queue [10,11]	*USE	*EXECUTE
	Output queue [7]	*READ	*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]	*USE	*EXECUTE
	New output queue, if changing the output queue[7]	*READ	*EXECUTE
	Current output queue, if changing the output queue	*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	Auxiliary storage pool (ASP) device description	*USE	
	Program library		*EXECUTE
DSPJOB [1]			
DSPJOBTBL			

Command	Referenced object	Authority needed	
		For object	For library
DSPJOBLOG [1,5]	Output file and member exist	*OBJOPR, *OBJMGT, *ADD	*EXECUTE
	Member does not exist	*OBJOPR, *OBJMGT, *ADD	*EXECUTE, *ADD
	Output file does not exist	*OBJOPR	*EXECUTE, *ADD
ENDGRPJOB			
ENDJOB [1]			
ENDJOBABN [1]			
ENDLOGSVR[6]			
ENDPJ [6]	Auxiliary storage pool (ASP) device description	*USE	
	Program library		*EXECUTE
HLDJOB [1]			
RLSJOB [1]			
RRTJOB			
RTVJOBA			
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, 17, 18]	Job description [9,11]	*USE	*EXECUTE
	Libraries in the library list (system, current, and user)[7]	*USE	
	Message queue[10]	*USE, *ADD	*EXECUTE
	User profile [10,11]	*USE	
	User profile in job description [10]	*USE (at level 40)	
	Job queue [10,11]	*USE	*EXECUTE
	Output queue [7]	*READ	*EXECUTE
	Sort sequence table [7]	*USE	*EXECUTE
	ASP devices in the initial ASP group	*USE	
SBMNETJOB	Database file	*USE	*EXECUTE
STRLOGSVR[6]			
STRPJ [6]	Subsystem description	*USE	
	Program	*USE	*EXECUTE
	Auxiliary storage pool (ASP) device description	*USE	

Command	Referenced object	Authority needed	
		For object	For library
TFRBCHJOB	Job queue	*READ	*EXECUTE
TFRGRPJOB	First group program	*USE	*EXECUTE
TFRJOB [8]	Job queue	*USE	*EXECUTE
	Subsystem description to which the job queue is allocated	*USE	
TFRSECJOB			
WRKACTJOB			
WRKARMJOB[16]			
WRKASPJOB	Device description	*USE	
WRKJOB [1]			
WRKJOBLOG			
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 a job log for a job that has all object (*ALLOBJ) special authority, you must have *ALLOBJ special authority or be authorized to the All Object Job Log function of the IBM i through Application Administration in IBM Navigator for i. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_ACCESS_ALLOBJ_JOBLOG, can also be used to change the list of users that are allowed to display a job log of a job with *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 workstation associated with the job must have a corresponding workstation entry in the subsystem description associated with the new subsystem. • The workstation 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 QPOZ*nnnnnn*, where '*nnnnnn*' is the job number of the job using the user trace facility.			
16 To work with a specific job or to display details of a specific job, one of the following conditions must apply: • The command must be issued from within that job. • The issuer of the command must be running under a user profile that is the same as the job user identity of the job. • The issuer of the command must be running under a user profile that has job control (*JOBCTL) special authority. **17** You must have the use (*USE) authority to the Changing Accounting Code (CHGACGCDE) command to specify a character-value accounting code on the Accounting code (ACGCDE) parameter. **18** You must have the job control (*JOBCTL) special authority to use the Submitted for (SBMFOR) parameter.			

Job description commands

This table lists the specific authorities required for the job description commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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, *READ	*EXECUTE
	User profile (USER)	*USE	
CRTJOBD (Q)	Job description		*READ, *ADD
	User profile (USER)	*USE	
DLTJOBD	Job description	*OBJEXIST	*EXECUTE
DSPJOBD	Job description	*OBJOPR, *READ	*EXECUTE
PRTJOBDAUT [1]			
WRKJOBD	Job description	Any	*USE

1
You must have *ALLOBJ or *AUDIT special authority to use this command.

Job queue commands

This table lists the specific authorities required for the job queue commands.

Command	Referenced object	Job queue parameters [4]		Special authority	Authority needed	
		AUTCHK	OPRCTL		For object	For library
CHGJOBQ	Job queue	*DTAAUT			*READ, *ADD, *DLT, *OBJMGMT	*EXECUTE
		*OWNER			Owner [2]	*EXECUTE
			*YES	*JOBCTL		*EXECUTE
CLRJOBQ [1]	Job queue	*DTAAUT			*READ, *ADD, *DLT	*EXECUTE
		*OWNER			Owner [2]	*EXECUTE
			*YES	*JOBCTL		*EXECUTE
CRTJOBQ [1]	Job queue					*READ, *ADD
DLTJOBQ	Job queue				*OBJEXIST	*EXECUTE
HLDJOBQ [1]	Job queue	*DTAAUT			*READ, *ADD, *DLT	*EXECUTE
		*OWNER			Owner [2]	*EXECUTE
			*YES	*JOBCTL		*EXECUTE
PRTQAUT [5]						

Command	Referenced object	Job queue parameters [4]		Special authority	Authority needed	
		AUTCHK	OPRCTL		For object	For library
RLSJOBQ [1]	Job queue	*DTAAUT			*READ, *ADD, *DLT	*EXECUTE
		*OWNER			Owner [2]	*EXECUTE
			*YES	*JOBCTL		*EXECUTE
WRKJOBQ [1,3]	Job queue	*DTAAUT			*READ	*EXECUTE
		*OWNER			Owner [2]	*EXECUTE
			*YES	*JOBCTL		*EXECUTE
WRKJOBQD	Job queue				*READ	*EXECUTE
			*YES	*JOBCTL		*EXECUTE

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 or *AUDIT special authority to use this command.

Job schedule commands

This table lists the specific authorities required for the 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
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

Command	Referenced object	Authority needed	
		For object	For library
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

This table lists the specific authorities required for the journal commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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
	Nonintegrated file system objects whose journaled changes are being applied	*OBJMGT, *CHANGE, *OBJEXIST	*EXECUTE, *ADD
	integrated file system objects whose journal changes are being applied	*RW, *OBJMGT	*RX (if subtree *ALL)
APYJRNCHGX (Q)	Journal	*USE	
	Journal receiver	*USE	
	File	*OBJMGT, *CHANGE, *OBJEXIST'	*EXECUTE, *ADD

Command	Referenced object	Authority needed	
		For object	For library or directory
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
CHGJRNA (Q) [10]			
CHGJRNOBJ [9]	Journal	*OBJOPR, *OBJMGT	
	Nonintegrated file system objects	*READ, *OBJMGT	
	Integrated file system objects	*R, *OBJMGT	*X
	Object path SUBTREE(*ALL)	*RX, *OBJMGT	
	Object path SUBTREE(*NONE)	*R, *OBJMGT	
CHGRMTJRN	Source journal	*CHANGE, *OBJMGT	*EXECUTE
	Source journal	*USE, *OBJMGT	*EXECUTE
CMPJRNIMG	Journal	*USE	*EXECUTE
	Journal receiver	*USE	*EXECUTE
	File	*USE	*EXECUTE
CPYAUDJRNE[8]	Output file already exists	*OBJOPR, *OBJMGT, *ADD, *DLT	*EXECUTE
	Output file does not exist		*EXECUTE, *ADD
CRTJRN	Journal		*READ, *ADD
	Journal receiver	*OBJOPR, *OBJMGT, *READ	*EXECUTE
DLTJRN	Journal	*OBJOPR, *OBJEXIST	*EXECUTE
DSPAUDJRNE[8]			

Command	Referenced object	Authority needed	
		For object	For library or directory
DSPJRN [6]	Journal	*USE	*EXECUTE
	Journal if FILE(*ALLFILE) is specified, no object selection is specified, the specified object has been deleted from the system, the specified object has never been journaled, *IGNFILSLT or *IGNOBJSLT is specified for any selected journal codes, or when OBJJID is specified, or the journal is a remote journal.	*OBJEXIST, *USE	*EXECUTE
	Journal receiver	*USE	*EXECUTE
	Nonintegrated file system object if specified	*USE	*EXECUTE
	Output file	Refer to the general rules.	Refer to the general rules.
	Integrated file system object if specified	*R (It can be *X as well if object is a directory and SUBTREE (*ALL) is specified)	*X
DSPJRNMNU[1]			
ENDJRN	See “Integrated file system commands” on page 439.		
ENDJRNAP	Journal	*OBJOPR, *OBJMGT	*EXECUTE
	File	*OBJOPR, *OBJMGT	*EXECUTE
ENDJRNLIB	Journal	*OBJOPR, *OBJMGT	*EXECUTE
	Library	*OBJOPR, *OBJMGT, *READ	
ENDJRNOBJ	Journal	*OBJOPR, *OBJMGT	*EXECUTE
	Object	*OBJOPR, *READ, *OBJMGT	*EXECUTE
ENDJRNPF	Journal	*OBJOPR, *OBJMGT	*EXECUTE
	File	*OBJOPR, *OBJMGT, *READ	*EXECUTE
JRNAP [2]			
JRNPF [3]			

Command	Referenced object	Authority needed	
		For object	For library or directory
RCVJRNE	Journal	*USE	*EXECUTE
	Journal if FILE(*ALLFILE) is specified, no object selection is specified, the specified object has been deleted from the system, the specified object has never been journaled, *IGNFILSLT or *IGNOBJSLT is specified for any selected journal codes, or when OBJJID is specified, or the journal is a remote journal.	*OBJEXIST, *USE	*EXECUTE
	Journal receiver	*USE	*EXECUTE
	Nonintegrated file system object if specified	*USE	*EXECUTE
	Integrated file system object if specified	*R (It can be *X as well if object is a directory and SUBTREE (*ALL) is specified)	*X
	Exit program	*EXECUTE	*EXECUTE
RMVJRNCHG (Q)	Journal	*USE	*EXECUTE
	Journal receiver	*USE	*EXECUTE
	Nonintegrated file system objects whose journaled changes are being removed	*OBJMGT, *CHANGE	*EXECUTE
RTVJRNE	Journal	*USE	*EXECUTE
	Journal if FILE(*ALLFILE) is specified, no object selection is specified, the specified object has been deleted from the system, the specified object has never been journaled, *IGNFILSLT or *IGNOBJSLT is specified for any selected journal codes, or when OBJJID is specified, or the journal is a remote journal.	*OBJEXIST, *USE	*EXECUTE
	Journal receiver	*USE	*EXECUTE
	Nonintegrated file system object if specified	*USE	*EXECUTE
	Integrated file system object if specified	*R (It can be *X as well if object is a directory and SUBTREE (*ALL) is specified)	*X
RMVRMTJRN	Source journal	*CHG, *OBJMGT	

Command	Referenced object	Authority needed	
		For object	For library or directory
SNDJRNE	Journal	*OBJOPR, *ADD	*EXECUTE
	Nonintegrated file system object if specified	*OBJOPR	*EXECUTE
	Integrated file system object if specified	*R	*X
STRJRN	See "Integrated file system commands" on page 439.		
STRJRNAP	Journal	*OBJOPR, *OBJMGT	*EXECUTE
	File	*OBJOPR, *OBJMGT	*EXECUTE
STRJRNLIB	Journal	*OBJOPR, *OBJMGT	*EXECUTE
	Library	*OBJOPR, *OBJMGT, *READ	
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	*USE	*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 or RST command.

5
*OBJOPR and *OBJEXIST authority is required for journal receivers if the option is chosen to delete receivers.

Command	Referenced object	Authority needed	
		For object	For library or directory
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 *AUDIT special authority to use this command. **9** To specify PTLTNS(*ALWUSE), you must have *ALLOBJ special authority. **10** You must have *JOBCTL special authority to use this command.			

Journal receiver commands

This table lists the specific authorities required for the 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 authority other than *EXECUTE is required for journal receivers if the option is chosen to display the description.			

Kerberos commands

This table lists the specific authorities required for the Kerberos commands.

Command	Referenced object	Object type	Authority needed for object
ADDKRBKTE	Each directory in the path name preceding the target key table file to be open.	*DIR	*X
	Parent directory of the target keytab file when add is specified, if the file does not already exist.	*DIR	*WX
	Keytab file when list is specified.	*STMF	*R
	Target keytab file when add or delete is specified.	*STMF	*RW
	Each directory in the path to the configuration files.	*DIR	*X
	Configuration files	*STMF	*R
ADDKRBTKT	Each directory in the path name preceding the key table file	*DIR	*X
	Key table file	*STMF	*R
	Each directory in the path name preceding the credentials cache file	*DIR	*X
	Credential cache file	*STMF	*RW
	Parent directory of the cache file to be used, if specified by the KRB5CCNAME environment variable, and the file is being created	*DIR	*WX
	Each directory in the path name to the configuration files	*DIR	*X
	Configuration files	*STMF	*R
CHGKRBPWD			
DLTKRBCCF	Each directory in the path name preceding the credentials cache file, if the credentials cache file does not reside in the default directory.	*DIR	*X
	Parent directory of the credentials cache file, if the credentials cache file does not reside in the default directory.	*DIR	*WX
	Credentials cache file, if the credentials cache file does not reside in the default directory.	*STMF	*RW, *OBJEXIST
	Each directory in the path name to the configuration files, if the credentials cache file does not reside in the default directory.	*DIR	*X
	Configuration files, if the credentials cache file does not reside in the default directory.	*STMF	*R

Command	Referenced object	Object type	Authority needed for object
DLTKRBCCF	All directories in the path name, if the credentials cache file resides in the default directory.	*DIR	*X
	Credentials cache file, if the credentials cache file resides in the default directory.	*STMF	*RW
	Each directory in the path to the configuration files, if the credentials cache file resides in the default directory.	*DIR	*X
	Configuration files, if the credentials cache file resides in the default directory.	*STMF	*R
DSPKRBCCF	Each directory in the path name preceding the key table file	*DIR	*X
	Key table file	*STMF	*R
	Each directory in the path name preceding the credentials cache file	*DIR	*X
	Credential cache file	*STMF	*RW
DSPKRBKTE	Each directory in the path name preceding the target key table file to be open.	*DIR	*X
	Parent directory of the target keytab file when add is specified, if the file does not already exist.	*DIR	*WX
	Keytab file when list is specified.	*STMF	*R
	Target keytab file when add or delete is specified.	*STMF	*RW
	Each directory in the path to the configuration files.	*DIR	*X
	Configuration files	*STMF	*R
RMVKRBKTE	Each directory in the path name preceding the target key table file to be open.	*DIR	*X
	Parent directory of the target keytab file when add is specified, if the file does not already exist.	*DIR	*WX
	Keytab file when list is specified.	*STMF	*R
	Target keytab file when add or delete is specified.	*STMF	*RW
	Each directory in the path to the configuration files.	*DIR	*X
	Configuration files	*STMF	*R

Language commands

This table lists the specific authorities required for the language commands.

Command	Referenced object	Authority needed	
		For object	For library
CLOSE	Close command	*USE	*EXECUTE
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)	Refer to the general rules.	*READ, *ADD
	Directory specified in OUTPUT, PPSRCSTMF or MAKEDEP parameter	*USE	*EXECUTE
	File specified in OUTPUT, PPSRCSTMF or MAKEDEP parameter	Refer to the general rules.	*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)	Refer to the general rules.	*READ, *ADD
	Binding directory	*USE	*EXECUTE
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
CRTBNDCL	Source file	*USE	*EXECUTE
	Include 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)	Refer to the general rules.	Refer to the general rules.
	Table specified in SRTSEQ parameter	*USE	*EXECUTE

Command	Referenced object	Authority needed	
		For object	For library
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)	Refer to the general rules.	*READ, *ADD
	Directory specified in OUTPUT, PPSRCSTMF, TEMPLATE or MAKEDEP parameter	*USE	*EXECUTE
	File specified in OUTPUT, PPSRCSTMF, TEMPLATE or MAKEDEP parameter	Refer to the general rules.	*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)	Refer to the general rules.	*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)	Refer to the general rules.	*READ, *ADD
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
CRTCLD	Source file	*USE	*EXECUTE
	Locale object - REPLACE(*NO)		*READ, *ADD
	Locale object - REPLACE(*YES)	Refer to the general rules.	*READ, *ADD

Command	Referenced object	Authority needed	
		For object	For library
CRTCLMOD	Source file	*USE	*EXECUTE
	Include 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)	Refer to the general rules.	Refer to the general rules.
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
CRTCLPGM	Source file	*USE	*EXECUTE
	Include 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)	Refer to the general rules.	Refer to the general rules.
	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)	Refer to the general rules.	*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)	Refer to the general rules.	*READ, *ADD
	File specified in OUTPUT, PPSRCSTMF or MAKEDEP parameter	*USE	*EXECUTE
	File specified in OUTPUT, PPSRCSTMF or MAKEDEP parameter	Refer to the general rules.	*READ, *ADD

Command	Referenced object	Authority needed	
		For object	For library
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)	Refer to the general rules.	*READ, *ADD
	Directory specified in OUTPUT, PPSRCSTMF, TEMPLATE or MAKEDEP parameter	*USE	*EXECUTE
	File specified in OUTPUT, PPSRCSTMF, TEMPLATE or MAKEDEP parameter	Refer to the general rules.	*READ, *ADD
	Headers generated by TEMPLATE parameter	*USE	*EXECUTE
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)	Refer to the general rules.	*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)	Refer to the general rules.	*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)	Refer to the general rules.	*READ, *ADD
	Source file for generated RPG program	Refer to the general rules.	Refer to the general rules.
	Externally described device files and database files referred to in source program	*OBJOPR	*EXECUTE
	Table specified in SRTSEQ parameter	*USE	*EXECUTE

Command	Referenced object	Authority needed	
		For object	For library
CRTS36CBL (S/36 environment)	Source file	*USE	*EXECUTE
	Program: REPLACE(*NO)		*READ, *ADD
	Program: REPLACE(*YES)	Refer to the general rules.	*READ, *ADD
CRTS36RPG	Source file	*USE	*READ, *ADD
	Program: REPLACE(*NO)		*READ, *ADD
	Program - REPLACE(*YES)	Refer to the general rules.	*READ, *ADD
CRTS36RPGR	Source file	*USE	*READ, *ADD
	Display file: REPLACE(*NO)		*READ, *ADD
	Display file: REPLACE(*YES)	Refer to the general rules.	*READ, *ADD
CRTS36RPT	Source file	*USE	*EXECUTE
	Source file for generated RPG program	Refer to the general rules.	Refer to the general rules.
	Program: REPLACE(*NO)		*READ, *ADD
	Program: REPLACE(*YES)	Refer to the general rules.	*READ, *ADD
CRTSQLCI (Db2 Query Manager and SQL Development for IBM i 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)	Refer to the general rules.	*READ, *ADD
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
CRTSQLCBL (Db2 Query Manager and SQL Development for IBM i 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)	Refer to the general rules.	*READ, *ADD
	Table specified in SRTSEQ parameter	*USE	*EXECUTE

Command	Referenced object	Authority needed	
		For object	For library
CRTSQLCBLI (Db2 Query Manager and SQL Development for IBM i 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)	Refer to the general rules.	*READ, *ADD
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
CRTSQLCPPI (Db2 Query Manager and SQL Development for IBM i 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)	Refer to the general rules.	*READ, *ADD
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
CRTSQLPLI (Db2 Query Manager and SQL Development for IBM i 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)	Refer to the general rules.	*READ, *ADD
	Table specified in SRTSEQ parameter	*USE	*EXECUTE
CRTSQLRPG (Db2 Query Manager and SQL Development for IBM i 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)	Refer to the general rules.	*READ, *ADD
	Table specified in SRTSEQ parameter	*USE	*EXECUTE

Command	Referenced object	Authority needed	
		For object	For library
CRTSQLRPGI (Db2 Query Manager and SQL Development for IBM i 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)	Refer to the general rules.	*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
DLTCLD	Locale object	*OBJEXIST, *OBJMGT	*EXECUTE
ENDCBLDBG (COBOL/400 licensed program or S/38 environment)	Program	*CHANGE	*EXECUTE
ENTCBLDBG (S/38 environment)	Program	*CHANGE	*EXECUTE
INCLUDE	Source file	*USE	*EXECUTE
RTVCLSRC	Program	*OBJMGT, *USE	*EXECUTE
	Service program	*OBJMGT, *USE	*EXECUTE
	Module	*OBJMGT, *USE	*EXECUTE
	Database source file	*OBJOPR, *OBJMGT, *ADD, *DLT	*EXECUTE
RTVCLDSRC	Locale object	*USE	*EXECUTE
	To-file	Refer to the general rules.	Refer to the general rules.
RUNSQLSTM[1]	Source file	*OBJOPR, *READ	*EXECUTE
STRCBLDBG	Program	*CHANGE	*EXECUTE
STRREXPRC	Source file	*USE	*EXECUTE
	Exit program	*USE	*EXECUTE

Command	Referenced object	Authority needed	
		For object	For library
STRSQL (Db2 Query Manager and SQL Development for IBM i licensed program) [1]	Sort sequence table	*USE	*EXECUTE
	Printer device description	*USE	*EXECUTE
	Printer output queue	*USE	*EXECUTE
	Printer file	*USE	*EXECUTE

1

See the Authorization, privileges and object ownership for more information about security requirements for structured query language (SQL) statements.

Library commands

This table lists the specific authorities required for the library commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, "Commands shipped with public authority *EXCLUDE," on page 355 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[14], *SBSD[14]	See the authority required by the DLT*xxx* command for the object type	
	ASP device (if specified)	*USE	
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]	ASP device (if specified)	*USE	

Command	Referenced object	Authority needed	
		For object	For library being acted on
DLTLIB [3]	Every object being deleted from library	*OBJEXIST	*USE, *OBJEXIST
	Object types *DTADCT[14], *JRN[14],*JRNRCV[14], *MSGQ, *SBSD[14]	See the authority required by the DLT*xxx* command for the object type	
	ASP device (if specified)	*USE	
DSPLIB	Library		*READ
	Objects in the library [5]	Some authority other than *EXCLUDE	
	ASP device (if specified)	*EXECUTE	
DSPLIBD	Library		Some authority other than *EXCLUDE
EDTLIBL	Library to add to list		*USE
RCLLIB	Library		*USE, *OBJEXIST
RSTLIB (Q)[7, 17, 19]	Media definition	*USE	*EXECUTE
	Library, if it does exist		*READ, *ADD
	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
	Output file, if specified	See General Rules	See General Rules
	QSYS/QASAVOBJ field reference file for output file, if an output file is specified and does not exist	*USE	*EXECUTE

Command	Referenced object	Authority needed	
		For object	For library being acted on
RSTLIB (Q)	Tape (QSYSTAP) or diskette (QSYSDKT) file	*USE [6]	*EXECUTE
	QSYS/QPSRLDSP printer output, if OUTPUT(*PRINT) specified	*USE	*EXECUTE
	Save file	*USE	*EXECUTE
	Optical File (OPTFILE)[12]	*R	Not applicable
	Path prefix of optical file (OPTFILE)[12]	*X	Not applicable
	Optical volume[11]	*USE	
	ASP device description[15]	*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
SAVLIB[18]	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	Refer to the general rules.	Refer to the general rules.
	QSYS/QASAVOBJ field reference file, if output file is specified and does not exist	*USE [6]	*EXECUTE
	QSYS/QPSAVOBJ printer output	*USE [6]	*EXECUTE
	Command user space, if specified	*USE	*EXECUTE

Command	Referenced object	Authority needed	
		For object	For library being acted on
SAVLIB	Optical File[12]	*RW	Not applicable
	Parent Directory of optical file (OPTFILE)[12]	*WX	Not applicable
	Path Prefix of optical file (OPTFILE)[12]	*X	Not applicable
	Root Directory (/) of Optical Volume[12, 13]	*RWX	Not applicable
	Optical volume[11]	*CHANGE	
	ASP device description[15]	*USE	
SAVRSTLIB	On the source system, same authority as required by SAVLIB command.		
	On the target system, same authority as required by RSTLIB command.		
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, 16, 20]	Library		*USE

Command	Referenced object	Authority needed	
		For object	For library being acted on
1 The authority needed for the library being acted on 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 a value other than *NONE for the Allow object differences (ALWOBJDIF) parameter. **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 the 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. **15** Authority required only if save or restore operation requires a library namespace switch.			

Command	Referenced object	Authority needed	
		For object	For library being acted on

16

This command requires *ALLOBJ special authority.

17

You must have *ALLOBJ special authority to specify *YES for the PVTAUT parameter.

18

You must have *ALLOBJ or *SAVSYS special authority to specify *YES for the PVTAUT parameter.

19

You must have *SAVSYS special authority to specify a name for the DFRID parameter.

20

If you are authorized to the IBM i Database Security Administrator function (QIBM_DB_SECADM) you do not need the specified authority to the object.

License key commands

This table lists the specific authorities required for the license key commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, "Commands shipped with public authority *EXCLUDE," on page 355 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	Refer to the general rules.	Refer to the general rules.
RMVLICKEY (Q)	Output file	*CHANGE	*EXECUTE

Licensed program commands

This table lists the specific authorities required for the licensed program commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, "Commands shipped with public authority *EXCLUDE," on page 355 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)			

Command	Referenced object	Authority needed	
		For Object	For Library

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

This table lists the specific authorities required for the 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
CHGLINETH [2]	Line description	*CHANGE, *OBJMGT	*EXECUTE
CHGLINPPP[2]	Line description	*CHANGE, *OBJMGT	*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
CRTLINETH [2]	Controller description (NETCTL)	*USE	*EXECUTE
	Line description		*READ, *ADD
	Network interface description (NWI)	*USE	*EXECUTE
	Network server description (NWS)	*USE	*EXECUTE
CRTLINPPP[2]	Controller description (NETCTL)	*USE	*EXECUTE
	Line description		*READ, *ADD
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

Command	Referenced object	Authority needed	
		For object	For library
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

This table lists the specific authorities required for the Local Area Network (LAN) commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, "Commands shipped with public authority *EXCLUDE," on page 355 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 CHGLANADPI	DSPLANADPP DSPLANSTS	RMVLANADPT (Q) RMVLANADPI	WRKLANADPT

Locale commands

This table lists the specific authorities required for the locale commands.

Command	Referenced object	Authority needed	
		For object	For library
CRTLOCALE	Source file	*USE	*USE, *ADD
DLTLOCALE	Locale	*OBJEXIST	*EXECUTE

Mail server framework commands

This table lists the specific authorities required for the mail server framework commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, "Commands shipped with public authority *EXCLUDE," on page 355 shows which IBM-supplied user profiles are authorized to the command. The security officer can grant *USE authority to others.

This command does not require any object authorities:			
ENDMSF (Q)	STRMSF (Q)		

Media commands

This table lists the specific authorities required for the media commands.

Command	Referenced object	Authority needed	
		For object	For library
ADDTAPCTG	Tape Library description	*USE	*EXECUTE
CFGDEVMLB[1]	Tape Library description	*CHANGE, *OBJMGT	*EXECUTE

Command	Referenced object	Authority needed	
		For object	For library
CHGDEVMLB (Q)	Tape Library description	*CHANGE, *OBJMGT	*EXECUTE
CHGJOBMLBA[4]	Tape Library description	*CHANGE	*EXECUTE
CHGTAPCTG	Tape Library description	*USE	*EXECUTE
CHKTAP	Tape device description	*USE	*EXECUTE
CRTTAPCGY	Tape Library description		
DLTMEDDFN	Media definition	*OBJEXIST	*EXECUTE
DLTTAPCGY	Tape Library description		
DMPTAP (Q)[5]	Tape device description	*USE	*EXECUTE
DSPTAP	Tape device description	*USE	*EXECUTE
DSPTAPCGY	Tape Library description		
DSPTAPCTG	Tape Library description	*USE	*EXECUTE
DSPTAPSTS	Tape Library description	*USE	*EXECUTE
DUPTAP	Tape device description	*USE	*EXECUTE
INZTAP	Tape device description	*USE	*EXECUTE
RMVTAPCTG	Tape Library 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.

3

To change the session media library attributes, you must have *CHANGE authority to the Tape Library description. To change the priority or work with another users job you must have *JOBCTL special authority.

4

To change the priority or work with another user's job you must have *JOBCTL special authority.

5

To use this command, you must have *ALLOBJ special authority when TYPE(*HEX) is specified or the tape has the secure volume flag or secured file flag set.

Menu and panel group commands

This table lists the specific authorities required for the menu and panel group commands.

Command	Referenced object	Authority needed	
		For object	For library
CHGMNU	Menu	*CHANGE	*USE

Command	Referenced object	Authority needed	
		For object	For library
CRTMNU	Source file	*USE	*EXECUTE
	Menu: REPLACE(*NO)		*READ, *ADD
	Menu: REPLACE(*YES)	Refer to the general rules.	*READ, *ADD
CRTPNLGRP	Panel group: Replace(*NO)		*READ, *ADD
	Panel group: REPLACE(*YES)	Refer to the general rules.	*READ, *ADD
	Source file	*USE	*EXECUTE
	Include file	*USE	*EXECUTE
CRTS36MNU	Menu: REPLACE(*NO)		*READ, *ADD
	Menu: REPLACE(*YES)	Refer to the general rules.	*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

This table lists the specific authorities required for the 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

This table lists the specific authorities required for the 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

Command	Referenced object	Authority needed	
		For object	For library
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

This table lists the specific authorities required for the 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 individual operations, you must have the authority required by the individual operation.

Message queue commands

This table lists the specific authorities required for the 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 individual operations, you must have the authority required by the individual operation.

Mode description commands

This table lists the specific authorities required for the mode description commands.

Command	Referenced object	Authority needed	
		For object	For library
CHGMODD [2]	Mode description	*CHANGE, *OBJMGT	*EXECUTE
CRTMODD [2]	Mode description		*EXECUTE
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

This table lists the specific authorities required for the 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 individual operations, you must have the authority required by the individual operation.

NetBIOS description commands

This table lists the specific authorities required for the 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

Command	Referenced object	Authority needed	
		For object	For library

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.

Network commands

This table lists the specific authorities required for the network commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, "Commands shipped with public authority *EXCLUDE," on page 355 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
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	Refer to the general rules.	Refer to the general rules.
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

Command	Referenced object	Authority needed	
		For object	For library

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 *IOSYSCFG, or *ALLOBJ and *IOSYSCFG special authorities.

Network file system commands

This table lists the specific authorities required for the network file system commands.

Command	Referenced object	Object type	File system	Authority needed for object
ADDMFS [1,3]	dir_to_be_ mounted_over	*DIR	"root" (/)	*W
CHGNFSEXP [1,2]	Path prefix	Refer to the general rules.		
DSPMFSINF	some_dirs	*DIR	"root" (/)	*RX
	Path prefix	Refer to the general rules.		
ENDNFSSVR [1,4]	none			
EXPORTFS [1,2]	Path prefix	Refer to the general rules.		
MOUNT [1,3]	dir_to_be_ mounted_over	*DIR	"root" (/)	*W
RLSIFSLCK [1]	object	*STMF	"root" (/), QOpenSys, UDFS	*R
	Path prefix	Refer to the general rules.		
RMVMFS [1]				
STATFS	some_dirs	*DIR	"root" (/)	*RX
	Path prefix	Refer to the general rules.		
STRNFSSVR [1]	none			
UNMOUNT [1]				

Command	Referenced object	Object type	File system	Authority needed for object
1 To use this command, you must have *IOSYSCFG special authority. **2** When the -F flag is specified and the /etc/exports file does not exist, you must have write, execute (*WX) authority to the /etc directory. When the -F flag is specified and the /etc/exports file does exist, you must have read, write (*RW) authority to the /etc/exports file and *X authority to the /etc directory. **3** The directory that is mounted over (dir_to_be_mounted_over) is any integrated file system directory that can be mounted over. **4** To end any daemon jobs started by someone else, you must have *JOBCTL special authority.				

Network interface description commands

This table lists the specific authorities required for the network interface description commands.

Command	Referenced object	Authority needed	
		For object	For library
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.

Network server commands

This table lists the specific authorities required for the 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	*STMF	"root" (/)	*RW
	Network server description	*NWSD	QSYS.LIB	*CHANGE, *OBJMGT
CHGNWSSTG[2]	Path (root and /QFPNWSSTG)	*DIR	"root" (/)	*WX
CHGNWSUSRA [4]	User Profile	*USRPRF		*OBJMGT, *USE
CRTNWSSTG [2]	Path (root and /QFPNWSSTG)	*DIR	"root" (/)	*WX

Command	Referenced object	Object type	File system	Authority needed for object
DLTINTSVR[5]	Network server description	*NWSD	QSYS.LIB	*OBJEXIST
	Line description	*LIND	QSYS.LIB	*OBJEXIST
	Network server storage space - Path (/QFPNWSSTG)	*DIR	"root" (/)	*WX
	Parent directory (name of the storage space)	*DIR	"root" (/)	*RWX, *OBJEXIST
	Files that make up the storage space	*STMF	"root" (/)	*OBJEXIST
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	*STMF	"root" (/)	*OBJEXIST
DLTWNTSVR[5]	Network server description	*NWSD	QSYS.LIB	*OBJEXIST
	Line description	*LIND	QSYS.LIB	*OBJEXIST
	Network server storage space - Path (/QFPNWSSTG)	*DIR	"root" (/)	*WX
	Parent directory (name of the storage space)	*DIR	"root" (/)	*RWX, *OBJEXIST
	Files that make up the storage space	*STMF	"root" (/)	*OBJEXIST
DSPNWSSTG	Path prefix	Refer to the general rules		
	Files that make up the storage space	*STMF	"root" (/)	*R
INSINTSVR[6]	Network server description	*NWSD	Not applicable	*USE
	Line description	*LIND	Not applicable	*USE
	Network server storage space - Path (/QFPNWSSTG)	*DIR	"root" (/)	*WX
INSWNTSVR[6, 7]	Network server description	*NWSD	Not applicable	*USE
	Line description	*LIND	Not applicable	*USE
	Network server configuration	*NWSCFG	Not applicable	*USE
	Network server storage space - Path (/QFPNWSSTG)	*DIR	"root" (/)	*WX

Command	Referenced object	Object type	File system	Authority needed for object
RMVNWSSTGL [2]	Path (/QFPNWSSTG)	*DIR	"root" (/)	*X
	Parent directory (name of the storage space)	*DIR	"root" (/)	*WX
	Files that make up the storage space	*STMF	"root" (/)	*RW
	Network server description	*NWSD	QSYS.LIB	*CHANGE, *OBJMGT
WRKNWSSTG	Path prefix	Refer to the general rules		
	Files that make up the storage space	*STMF	"root" (/)	*R
These commands do not require any object authorities:				
ADDRMTSVR CHGNWSA [4](Q) CHGNWSALS CRTNWSALS DLTNWSALS DSPNWSA	DSPNWSALS DSPNWSSSN DSPNWSSTC DSPNWSUSRA SBMNWSCMD (Q) [3]		SNDNWSMSG WRKNWSALS WRKNWSENR WRKNWSSSN WRKNWSSTS	

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 parameters.

5 To use this command, you must have *IOSYSCFG and *ALLOBJ special authorities.

6 To use this command, you must have *IOSYSCFG, *ALLOBJ, and *JOBCTL special authorities.

7 You must have *SECADM special authority to specify a nondefault value for the IPSECRULE, CHAPAUT, or SPCERTID parameter.

Network server configuration commands

This table lists the specific authorities required for the network server configuration commands.

Command	Referenced object	Authority needed	
		For object	For QUSRSYS library
CHGNWSCFG[1, 3]	Network server configuration	*CHANGE	*EXECUTE
CRTNWSCFG [1, 3]	Network server configuration	*USE	*READ, *ADD

Command	Referenced object	Authority needed	
		For object	For QUSRSYS library
DLTNWSCFG[1, 3]	Network server configuration	*OBJEXIST	*EXECUTE
DSPNWSCFG[1, 3]	Network server configuration	*USE	*EXECUTE
INZNWSCFG[1, 2]	Network server configuration	*CHANGE	*EXECUTE
WRKNWSCFG[1]	Network server configuration	*USE	*EXECUTE

1
To use this command, you must have *IOSYSCFG special authority.

2
To use this command, you must have *SECADM special authority.

3
To specify or view a nondefault value for the IPSECRULE, CHAPAUT, or SPCERTID parameter, you must have security administrator (*SECADM) special authority.

Network server description commands

This table lists the specific authorities required for the 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

This table lists the specific authorities required for the node list commands.

Command	Referenced object	Authority needed	
		For object	For library
ADDNODLE	Node list	*OBJOPR, *ADD	*EXECUTE
CRTNODL	Node list		*READ, *ADD

Command	Referenced object	Authority needed	
		For object	For library
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

This table lists the specific authorities required for the office services commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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.			
ADDACC (Q) DSPACC DSPACCAUT DSPUSRPMN	GRTACCAUT [2,3,6] (Q) GRTUSRPMN [1,2] RMVACC [1] (Q) RVKACCAUT [1]	RVKUSRPMN [1,2] WRKDOCLIB [4] WRKDOCPRTQ [5]	

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 mails 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.

6

You must have all object (*ALLOBJ) or security administrator (*SECADM) special authority to grant access code authority for other users.

Online education commands

This table lists the specific authorities required for the online education commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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
CVTEDU			
STREDU			

Operational assistant commands

This table lists the specific authorities required for the operational assistant commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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
CHGBCKUP [1]	QUSRSYS/QEZBACKUPL *USRIDX	*CHANGE	*EXECUTE
CHGCLNUP[2]			
CHGPWRSCD [3]			
CHGPWRSCDE [3]			
DSPBCKSTS	QUSRSYS/QEZBACKUPL *USRIDX	*USE	*EXECUTE
DSPBCKUP	QUSRSYS/QEZBACKUPL *USRIDX	*USE	*EXECUTE
DSPBCKUPL	QUSRSYS/QEZBACKUPL *USRIDX	*USE	*EXECUTE
	QUSRSYS/QEZBACKUPF *USRIDX	*USE	*EXECUTE
DSPPWRSCD			
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
	ASP device (if specified)	*USE	
RTVBCKUP	QUSRSYS/QEZBACKUPL *USRIDX	*USE	*EXECUTE
RTVCLNUP			
RTVDSKINF (Q) [5]	ASP device (if specified)	*USE	
RTVPWRSCDE	DSPPWRSCD command	*USE	
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

Command	Referenced object	Authority needed	
		For object	For library

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

This table lists the specific authorities required for the optical commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, "Commands shipped with public authority *EXCLUDE," on page 355 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		
		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	Not applicable	Not applicable
	Optical Device	*USE	*EXECUTE	*CHANGE[3]
	Server CSI	*USE	*EXECUTE	Not applicable
CHKOPTVOL	Optical device	*USE	*EXECUTE	*USE
	Root directory (/) of volume	*RWX	Not applicable	Not applicable

Command	Referenced object	Authority needed		
		Object	**Library**	**Optical volume [1]**
CPYOPT	Optical Device	*USE	*EXECUTE	*USE - Source Volume
				*ALL - Target Volume
	Each preceding dir in path of source file	*X	Not applicable	Not applicable
	Each preceding dir in path of destination file	*X	Not applicable	Not applicable
	Source file (*DSTMF)[5]	*R	Not applicable	Not applicable
	Parent dir of destination file	*WX	Not applicable	Not applicable
	Parent of parent dir if creating dir	*WX	Not applicable	Not applicable
CPYOPT	Destination file if replaced due to SLTFILE(*ALL)	*W	Not applicable	Not applicable
	Destination file if replaced due to SLTFILE(*CHANGED)	*RW	Not applicable	Not applicable
	Each dir in path that precedes source dir	*X	Not applicable	Not applicable
	Each dir in path that precedes target dir	*X	Not applicable	Not applicable
CPYOPT	Dir being copied[5]	*R	Not applicable	Not applicable
	Dir being copied if it contains entries	*RX	Not applicable	Not applicable
	Parent of target dir	*WX	Not applicable	Not applicable
	Target dir if replaced due to SLTFILE(*ALL)	*W	Not applicable	Not applicable
	Target dir if replaced due to SLTFILE(*CHANGED)	*RW	Not applicable	Not applicable
	Target dir if entries are to be created	*WX	Not applicable	Not applicable
CPYOPT	Source files	*R	Not applicable	Not applicable
	Destination file if replaced due to SLTFILE(*ALL)	*W	Not applicable	Not applicable
	Destination file if replaced due to SLTFILE(*CHANGED)	*RW	Not applicable	Not applicable
CRTDEVOPT[4]	Optical Device		*EXECUTE	

Command	Referenced object	Authority needed		
		Object	Library	Optical volume [1]
CVTOPTBKU	Optical Device	*USE	*EXECUTE	*ALL
DSPOPT	Path Prefix when DATA (*SAVRST)[5]	*X	Not applicable	Not applicable
	File Prefix when (*SAVRST)[2]	*R	Not applicable	Not applicable
	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	Not applicable	Not applicable
	Optical Device	*USE	*EXECUTE	*ALL
LODOPTFMW	Stream file	*R	Not applicable	Not applicable
	Path prefix	Refer to the general rules.		
RCLOPT (Q)	Optical Device	*USE	*EXECUTE	
RMVOPTCTG (Q)	Optical Device	*USE	*EXECUTE	
RMVOPTSVR (Q)	Server CSI	*USE	*EXECUTE	
STRNETINS (Q)[6]	Network optical device	*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	

Command	Referenced object	Authority needed		
		Object	Library	Optical volume [1]

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).

6

You must have *JOBCTL special authority to use this command.

Output queue commands

This table lists the specific authorities required for the output queue commands.

Command	Referenced object	Output queue parameters		Special authority	Authority needed	
		AUTCHK	OPRCTL		For object	For library
CHGOUTQ [1]	Data queue				*READ	*EXECUTE
	Output queue	*DTAAUT			*OBJMGT, *READ, *ADD, *DLT	*EXECUTE
		*OWNER			Owner [2]	*EXECUTE
			*YES	*JOBCTL		*EXECUTE
	Message queue				*OBJOPR *ADD	*EXECUTE
	Workstation customization object				*USE	*EXECUTE
	User-data transform program				*OBJOPR *EXECUTE	*EXECUTE
	User-driver program				*OBJOPR *EXECUTE	*EXECUTE
CLROUTQ [1]	Output queue	*DTAAUT			*READ, *ADD, *DLT	*EXECUTE
		*OWNER			Owner [2]	*EXECUTE
			*YES	*JOBCTL		*EXECUTE
CRTOUTQ	Data queue				*READ	*EXECUTE
	Output queue					*READ, *ADD
	Message queue				*OBJOPR *ADD	*EXECUTE
	Workstation customization object				*USE	*EXECUTE
DLTOUTQ	Output queue				*OBJEXIST	*EXECUTE
HLDOUTQ [1]	Output queue	*DTAAUT			*READ, *ADD, *DLT	*EXECUTE
		*OWNER			Owner [2]	*EXECUTE
			*YES	*JOBCTL		*EXECUTE
PRTQAUT [4]						
RLSOUTQ [1]	Output queue	*DTAAUT			*READ, *ADD, *DLT	*EXECUTE
		*OWNER			Owner [2]	*EXECUTE
			*YES	*JOBCTL		*EXECUTE

Command	Referenced object	Output queue parameters		Special authority	Authority needed	
		AUTCHK	OPRCTL		For object	For library
WRKOUTQ [1,3]	Output queue				*READ	*EXECUTE
			*YES	*JOBCTL		*EXECUTE
WRKOUTQD [1,3]	Output queue				*READ	*EXECUTE
			*YES	*JOBCTL		*EXECUTE

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

This table lists the specific authorities required for the 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

This table lists the specific authorities required for the performance commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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
ADDDWDFN (Q) [7]			

Command	Referenced object	Authority needed	
		For object	For library
ADDJWDFN (Q) [7]			
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	
ANZCMDPFR (Q)	Command file	*USE	*EXECUTE
	Output file	*USE	*EXECUTE, *ADD
ANZDBF (Q)	QPFR/QPTANZDC *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
ANZPFRDTA (Q)	QPFR/QAVCPP *PGM	*USE	*EXECUTE
	Performance data[2]		*ADD, *READ
ANZPFRDT2 (Q)	QPFR/QAVCPP *PGM	*USE	*EXECUTE
	QAPTAPGP *FILE	*CHANGE	*EXECUTE
	DLTFCNARA command (Q)	*USE	*EXECUTE
	QPFR/QPTAGRP *PGM	*USE	*EXECUTE
CFGPFRCOL (Q)	Collection library		*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
CHGMGTCOL	MGTCOL	*OBJMGT	
	User library		*EXECUTE
CHGPEXDFN (Q)[5]	PGM library		*EXECUTE
CHKPFRCOL (Q)			

Command	Referenced object	Authority needed	
		For object	For library
CPYFCNARA (Q)	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)	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
CPYPFRCOL (Q)	From library		*EXECUTE
	To library		*EXECUTE, *ADD
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
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

Command	Referenced object	Authority needed	
		For object	For library
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
CRTPFRSUM (Q)	User library		*ADD, *READ
CVTPFRCOL (Q)	From library		*USE
	To library		*USE, *ADD
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
DLTFCNARA (Q)	QPFR/QPTAGRPD *PGM	*USE	*EXECUTE
	QAPGGPHF *FILE in the graph format library	*CHANGE	*EXECUTE
DLTGPHFMT (Q)	QPFR/QPGDLTGP *PGM	*USE	*EXECUTE
	QAPGPKGF *FILE in the graph package library	*CHANGE	*EXECUTE
DLTGPHPKG (Q)	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

Command	Referenced object	Authority needed	
		For object	For library
DLTHSTDTA (Q)	QPFR/QPGDLTHS *PGM	*USE	*EXECUTE
DLTPEXDTA (Q)[5]	Data Library [1]		*EXECUTE, *DELETE [2]
DLTPFRCOL (Q)	Library		*EXECUTE
DLTPFRDTA (Q)	QPFR/QPTDLTCP *PGM	*USE	*EXECUTE
DMPMEMINF	Output file	Refer to the general rules	Refer to the general rules
DMPTRC (Q)[5]	Library where the trace data will be stored		*EXECUTE, *ADD
	Output file (QAPTPAGD)	*CHANGE	*EXECUTE, *ADD
DSPHSTGPH (Q)	QPFR/QPGCTRL *PGM	*USE	*EXECUTE
	Historical data library		*EXECUTE
DSPPFRDTA (Q)	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)	QPFR/QPGCTRL *PGM	*USE	*EXECUTE
	Output file library		*EXECUTE
	Job description	*USE	*EXECUTE
ENDDW (Q)[7]			
ENDJOBTRC (Q)	QPFR/QPTTRCJ0 *PGM	*USE	*EXECUTE
ENDJW (Q)[7]			
ENDPEX (Q)[5]	Data Library [1]		*READ, *ADD [2]
ENDPFRCOL (Q)			
MOVPFRCOL (Q)	From library		*EXECUTE
	To library		*EXECUTE, *ADD
PRTACTRPT (Q)	QPFR/QITPRTAC *PGM	*USE	*EXECUTE
	Performance data[2]	*USE	*ADD, *READ
	Job description	*USE	*EXECUTE
PRTCPTRPT (Q)	QPFR/QPTCPTRP *PGM	*USE	*EXECUTE
	Performance data[2]		*ADD, *READ
	Job description	*USE	*EXECUTE

Command	Referenced object	Authority needed	
		For object	For library
PRTJOBRPT (Q)	QPFR/QPTITVXC *PGM	*USE	*EXECUTE
	Performance data[2]		*ADD, *READ
	Job description	*USE	*EXECUTE
PRTJOBTRC (Q)	QPFR/QPTTRCRP *PGM	*USE	*EXECUTE
	Job trace file (QAPTTRCJ) library		*EXECUTE
	Job description	*USE	*EXECUTE
PRTLCKRPT (Q)	QPFR/QPTLCKQ *PGM	*USE	*EXECUTE
PRTPEXRPT[5]	Data Library[1]		*EXECUTE[2]
	Output file	*USE	*EXECUTE, *ADD
	QPFR/QVPEPRTC *PGM	*USE	*EXECUTE
	QPFR/QVPESVGN *SRVPGM	*USE	*EXECUTE
	QPFR/QYPESVGN *SRVPGM	*USE	*EXECUTE
PRTPOLRPT (Q)	QPFR/QPTITVXC *PGM	*USE	*EXECUTE
	Performance data[2]		*ADD, *READ
	Job description	*USE	*EXECUTE
PRTRSCRPT (Q)	QPFR/QPTITVXC *PGM	*USE	*EXECUTE
	Performance data[2]		*ADD, *READ
	Job description	*USE	*EXECUTE
PRTSYSRPT (Q)	QPFR/QPTSYSRP *PGM	*USE	*EXECUTE
	QAPMDMPT *FILE		*EXECUTE
	Job description	*USE	*EXECUTE
PRTTNSRPT (Q)	QPFR/QPTTNSRP *PGM	*USE	*EXECUTE
	Trace file (QTRJOBT) library		*EXECUTE
	Job description	*USE	*EXECUTE
PRTTRCRPT (Q)	QPFR/QPTTRCCP *PGM	*USE	*EXECUTE
RMVDWDFN (Q) [7]			
RMVJWDFN (Q) [7]			
RMVPEXDFN (Q)[5]			
RMVPEXFTR (Q)[5]			
RSTPFRCOL (Q)	Library associated with the restore collection	*EXECUTE, *ADD [6]	
	Save file	*USE	*EXECUTE
	ASP device (if specified)	*USE	

Command	Referenced object	Authority needed	
		For object	For library
SAVPFRCOL (Q)	Library containing collection to be saved	*EXECUTE [6]	
	Save file, if empty	*USE, *ADD	*EXECUTE, *ADD
	Save file, if records exist in it	*OBJMGT, *USE, *ADD	*EXECUTE
STRDBMON[3]	Output file	*OBJOPR, *ADD	*EXECUTE
STRDW (Q)[7]	User library		*EXECUTE
STRJOBTRC (Q)	QPFR/QPTTRCJ1 *PGM	*USE	*EXECUTE
STRJW (Q) [7]	User library		*EXECUTE
STRPEX (Q)[5]			
STRPFRCOL (Q)			
STRPFRG (Q)	QPFR/QPGSTART *PGM	*USE	*EXECUTE
STRPFRT (Q)	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)	QPFR/QPTAGRPC *PGM	*USE	*EXECUTE
	Output file (QAITMON)	*CHANGE, *ALTER	*EXECUTE, *ADD
WRKPEXDFN (Q)[5]			
WRKPEXFTR (Q)[5]			
WRKSYSACT (Q)[3]	QPFR/QITMONCP *PGM	*USE	*EXECUTE
These commands do not require any object authorities: • ENDDBMON[3] • ENDPFRTRC (Q) • STRPFRTRC (Q)			

Command	Referenced object	Authority needed	
		For object	For library

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 the STRDBMON or ENDDBMON commands, where the JOB command parameter uses a generic name or a specific name which belongs to a user which is different from the current user, requires that you have *JOBCTL special authority or be authorized to the SQL Administrator function of IBM i through Application Administration in IBM Navigator for i. The Change Function Usage Information (CHGFCNUSG) command, with a function ID of QIBM_DB_SQLADM, can also be used to change the list of authorized users.

5
To use this command, you must have *SERVICE special authority or you must be authorized to the Service Trace function of IBM i through Application Administration in IBM Navigator for i. The Change Function Usage (CHGFCNUSG) command, 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.

6
If you have *SAVSYS special authority, you do not need the authority specified.

7
To use this command, you must have service (*SERVICE) special authority, or be authorized to the Disk Watcher function of the operating system through IBM Navigator for i Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_DISK_WATCHER, can also be used to change the list of users that are allowed to use the disk watcher tool.

Print descriptor group commands

This table lists the specific authorities required for the 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

This table lists the specific authorities required for the print services facility configuration commands.

Command	Referenced object	Authority needed	
		For object	For library
CHGPSFCFG [1, 2]			
CRTGPSFCFG [1, 2]			*READ, *ADD

Command	Referenced object	Authority needed	
		For object	For library
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 authority is required to use this command.

Problem commands

This table lists the specific authorities required for the problem commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, "Commands shipped with public authority *EXCLUDE," on page 355 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	Refer to the general rules.	Refer to the general rules.
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[4] (Q)	Device description	*USE, *OBJMGT	*EXECUTE
VFYPRT (Q)	Device description	*USE	*EXECUTE
WRKPRB (Q) [2]	Line, controller, NWID (Network ID), and device based on problem analysis action	*USE	*EXECUTE

Command	Referenced object	Authority needed	
		For object	For library

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.

4
You must have *IOSYSCFG special authority when the device description is allocated by a media library device.

Program commands

This table lists the specific authorities required for the program commands.

Command	Referenced object	Authority needed	
		For object	For library
The object authorities required for the CRT*xxx* PGM commands are listed in the Languages table in "Language commands" on page 478.			
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 re-create 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 re-create 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

Command	Referenced object	Authority needed	
		For object	For library
CLRTRCDTA [1]			
CRTPGM	Program, Replace(*NO)	Refer to the general rules.	*READ, *ADD
	Program, Replace(*YES)	Refer to the general rules.	*READ, *ADD
	Service program specified in the BNDSRVPGM parameter.	*USE	*EXECUTE
	Module	*USE	*EXECUTE
	Binding directory	*USE	*EXECUTE
CRTSRVPGM	Service program, Replace(*NO)	Refer to the general rules.	*READ, *ADD
	Service program, Replace(*YES)	Refer to the general rules.	*READ, *ADD
	Module	*USE	*EXECUTE
	Service program specified in BNDSRVPGM parameter	*USE	*EXECUTE
	Export source file	*OBJOPR *READ	*EXECUTE
	Binding directory	*USE	*EXECUTE
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	Refer to the general rules.	Refer to the general rules.

Command	Referenced object	Authority needed	
		For object	For library
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]			
RTVCLSRC	Program	*OBJMGT, *USE	*EXECUTE
	Service program	*OBJMGT, *USE	*EXECUTE
	Module	*OBJMGT, *USE	*EXECUTE
	Database source file	*OBJOPR, *OBJMGT, *ADD, *DLT	*EXECUTE
SETATNPGM	Attention-key-handling program	*EXECUTE	*EXECUTE
SETPGMINF	Database files	*OBJOPR	*EXECUTE
	Source file	*USE	*EXECUTE
	Root program	*CHANGE	*READ, *ADD
	Subprogram	*USE	*EXECUTE
STRCBLDBG	Program	*CHANGE	*EXECUTE

Command	Referenced object	Authority needed	
		For object	For library
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

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 See the Authorization, privileges and object ownership for more information about security requirements for SQL statements.

Command	Referenced object	Authority needed	
		For object	For library
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.			

QSH shell interpreter commands

This table lists the specific authorities required for the QSH shell interpreter commands.

The commands listed in this table do not require any authorities to objects.

Command	Referenced object	Authority needed	
		For object	For library
STRQSH [1, 2]			
QSH[1, 2]			
1 QSH is an alias for the STRQSH CL command. **2** You need *RX authority to all scripts and *X authority to all directories in the path to the script.			

Query commands

This table lists the specific authorities required for the 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

Command	Referenced object	Authority needed	
		For object	For library
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
RUNQRY	Query definition	*USE	*USE
	Input files	*USE	*EXECUTE
	Output files	Refer to the general rules.	Refer to the general rules.
STRQMQRY [1]	Query management query	*USE	*EXECUTE
	Query management form, if specified	*USE	*EXECUTE
	Query definition, if specified	*USE	*EXECUTE
	Output file	Refer to the general rules.	Refer to the general rules.
	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]			

Command	Referenced object	Authority needed	
		For object	For library

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 the CHGQRYA command, you must have *JOBCTL special authority or be authorized to the SQL Administrator function of IBM i through Application Administration in IBM Navigator for i. The Change Function Usage Information (CHGFCNUSG) command, with a function ID of QIBM_DB_SQLADM, can also be used to change the list of authorized users.

Question and answer commands

This table lists the specific authorities required for the question and answer commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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 QAQA*xx*BQPY [1]	*READ	*READ
ASKQST	Database file QAQA*xx*BBPY [1] or QAQA*xx*BQPY [1]	*READ	*READ
CHGQSTDB (Q)	Database file QAQA*xx*BQPY [1]	*READ	*READ
CRTQSTDB [2] (Q)	Database files		*READ, *ADD, *EXECUTE
CRTQSTLOD (Q)	Database file QAQA*xx*BQPY [1]	*READ	*READ
DLTQST (Q)	Database file QAQA*xx*BQPY [1]	*READ	*READ
DLTQSTDB (Q)	Database file QAQA*xx*BQPY [1]	*READ	*READ
EDTQST (Q)	Database file QAQA*xx*BQPY [1]	*READ	*READ
LODQSTDB [2] (Q)	Database file QAQA*xx*BQPY [1,3]	*READ	*READ, *ADD, *EXECUTE
STRQST [4]	Database file QAQA*xx*BBPY [1] or QAQA*xx*BQPY [1]	*READ	*READ
WRKQST	Database file QAQA*xx*BBPY [1] QAQA*xx*BQPY [1]	*READ	*USE
WRKCNTINF			*EXECUTE

Command	Referenced object	Authority needed	
		For object	For library

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 QAQA*xx*BBPY, is set to *EXCLUDE. Public authority for QAQA*xx*BBPY 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

This table lists the specific authorities required for the 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 do 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

This table lists the specific authorities required for the registration facility commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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
ADDEXITPGM (Q)			
RMVEXITPGM (Q)			

Command	Referenced object	Authority needed	
		For object	For library
WRKREGINF			

Relational database commands

This table lists the specific authorities required for the 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 device description[7]	*CHANGE	
DSPRDBDIRE	Output file, if specified	Refer to the general rules.	Refer to the general rules.
These commands do not require any authority to objects:			
RMVRDBDIRE WRKRDBDIRE			

1
Authority verified when the RDB directory entry is used.

Resource commands

This table lists the specific authorities required for the resource commands.

Command	Referenced object	Authority needed	
		For object	For library
DSPHDWRSC			
DSPSFWRSC	Output file, if specified	Refer to the general rules.	Refer to the general rules.
EDTDEVRSC			
WRKHDWRSC [1]			

1
If you use the option to create a configuration object, you must have authority to use the appropriate CRT command.

Remote Job Entry (RJE) commands

This table lists the specific authorities required for the Remote Job Entry (RJE) 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

Command	Referenced object	Authority needed	
		For object	For library
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
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

Command	Referenced object	Authority needed	
		For object	For library
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
CRTRJECFG	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

Command	Referenced object	Authority needed	
		For object	For library
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
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

Command	Referenced object	Authority needed	
		For object	For library
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
Review the authorities that are required for the SBMJOB command.

8
To use an individual operation, you must have the authority required by the operation.

Security attributes commands

This table lists the specific authorities required for the security attributes commands.

Command	Referenced object	Authority needed	
		For object	For library
CHGSECA [1]			

Command	Referenced object	Authority needed	
		For object	For library
CHGSECAUD [2,3]			
CFGSYSSEC [1,2,3]			
DSPSECA			
DSPSECAUD [3]			
PRTSYSSECA [4]			

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.

4
You must have *ALLOBJ or *AUDIT special authority to use this command.

Server authentication entry commands

This table lists the specific authorities required for the 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

This table lists the specific authorities required for the service commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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
ADDTRCFTR[11]			
APYPTF (Q)	Product library	*OBJMGT	
CHGSRVA [3] (Q)			
CHKCMNTRC [3] (Q)			*EXECUTE

Command	Referenced object	Authority needed	
		For object	For library
CHKPRDOPT (Q)	All objects in product option [4]		
CPYFRMMSD [12] (Q)	Stream file, if it already exists	*W	
	Stream file path name prefix	*X	
	Stream file parent directory, if the stream file does not exist	*WX	
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
CPYTOMSD [12] (Q)	Stream file	*R	
	Stream file path name prefix	*X	
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	Refer to the general rules.	Refer to the general rules.
DSPPTFAPYI (Q)	Output file	Refer to the general rules.	Refer to the general rules.
DSPPTFGRP (Q)			

Command	Referenced object	Authority needed	
		For object	For library
DSPSRVA (Q)			
DSPSRVSTS (Q)			
DSPSSTUSR [19]			
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	
ENDWCH[16] (Q)	Watch sessions watching for a message within a job log[17]		
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	Refer to the general rules.	Refer to the general rules.
PRTERRLOG (Q)	Output file	Refer to the general rules.	Refer to the general rules.
PRTINTDTA[12,13] (Q)			
PRTTRC[11](Q)	QSYS Library	*EXECUTE	
	Database Files	*USE	
	DLTTRC command	*USE	
RCLAPPN[20](Q)	Controller description	*USE, *OBJMGT	
	Device description	*USE, *OBJMGT	
RMVPTF (Q)	Product library	*OBJMGT	
RMVTRCFTR[11]			
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

Command	Referenced object	Authority needed	
		For object	For library
SNDPTFORD [10] (Q)	CRTIMGCLG	*USE	
	QUSRSYS		*ADD, *READ
SNDSRVRQS (Q)			
STRCMNTRC [11] (Q)	NWID (network ID) or line description	*USE	*EXECUTE
	Watched job[17]		
	Trace exit program	*OBJOPR and *EXECUTE	*EXECUTE
	Message queue	*USE	*USE
STRCPYSCN	Job queue	*USE	*EXECUTE
	Device description	*USE	*EXECUTE
	Output file, if specified	Refer to the general rules.	Refer to the general rules.
STRSRVJOB (Q)	User profile of job	*USE	*EXECUTE
STRSST [3] (Q)			
STRTRC (Q)[11, 15]	Watched job[17]		
	Trace exit program	*OBJOPR and *EXECUTE	*EXECUTE
	Message queue	*USE	*USE
STRWCH[16] (Q)	Watched job[17]		
	Watch exit program	*OBJOPR and *EXECUTE	*EXECUTE
	Message queue	*USE	*USE
TRCCNN[11](Q)	Watched job[17]		
	Trace exit program	*OBJOPR and *EXECUTE	*EXECUTE
	Message queue	*USE	*USE
TRCCPIC (Q)			
TRCICF (Q)			
TRCINT[11] (Q)	Watched job[17]		
	Trace exit program	*OBJOPR and *EXECUTE	*EXECUTE
	Message queue	*USE	*USE
TRCJOB (Q)	Output file, if specified	Refer to the general rules.	Refer to the general rules.
	Exit program, if specified	*USE	*EXECUTE

Command	Referenced object	Authority needed	
		For object	For library
TRCTCPAPP[11] (Q)	Line description	*USE	
	Network interface	*USE	
	Network interface	*USE	
	Watched job[17]		
	Trace exit program	*OBJOPR and *EXECUTE	*EXECUTE
	Message queue	*USE	*USE
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[14] (Q)	Device description	*USE, *OBJMGT	*EXECUTE
WRKCNTINF (Q)			
WRKFSTAF (Q)	QUSRSYS/QPVINDEX *USRIDX	*CHANGE	*USE
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)			
WRKPTFORD (Q)	QESCPTFO and SNDPTFORD	*USE	
WRKSRVPVD (Q)			
WRKTRC[11] (Q)			
WRKWCH[18] (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
You must have Service (*SERVICE) special authority to use 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.

Command	Referenced object	Authority needed	
		For object	For library
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. The following restrictions apply for the IMGDIR parameter: • You must have *X authority to each directory in the path. • You must have *WX authority to the directory that contains optical image.			
11 To use this command, you must have *SERVICE special authority, or be authorized to the Service Trace function of IBM i through Application Administration in IBM Navigator for i. The Change Function Usage Information (CHGFCNUSG) command, 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. **12** To use this command, you must have *SERVICE special authority, or be authorized to the Service Dump Function of IBM i through Application Administration in IBM Navigator for i. The Change Function Usage Information (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_DUMP, can also be used to change the list of users that are allowed to perform dump operations. **13** This command must be issued from within the job with internal data being printed, or the issuer of the command must be running under a user profile which is the same as the job user identity of the job with internal data being printed, or the issuer of the command must be running under a user profile which has job control (*JOBCTL) special authority. **14** You must have *IOSYSCFG special authority when the device description is allocated by a media library device. **15** If you specify a generic user name for the Job name (JOB) parameter, you must have all object (*ALLOBJ) special authority, or be authorized to the Trace Any User function of IBM i through Application Administration in IBM Navigator for i. You can also use the Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_ALLOBJ_TRACE_ANY_USER, to change the list of users that are allowed to perform trace operations.		

Command	Referenced object	Authority needed	
		For object	For library

16

To use this command, you must have service (*SERVICE) special authority, or be authorized to the service watch function of IBM i through Application Administration in IBM Navigator for i. You can also use the Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_WATCH, to change the list of users that are allowed to start and end watch operations.

17

Job control (*JOBCTL) special authority is needed if the job is running under a different user from the job user identity of the job being watched. All object (*ALLOBJ) special authority is needed if *ALL is specified for the watched job name, or if a generic user name is specified. A user that does not have *ALLOBJ special authority can perform the function if they are authorized to the Watch Any Job function of IBM i through Application Administration in IBM Navigator for i. You can also use the Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_WATCH_ANY_JOB, to change the list of users that are allowed to start and end watch operations.

18

To use this command, you must have service (*SERVICE) special authority, or be authorized to the service trace function and service watch function of IBM i through Application Administration in IBM Navigator for i. You can also use the Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_TRACE and QIBM_SERVICE_WATCH, to change the list of users that are allowed to perform trace operations.

19

You must have Audit (*AUDIT) and Security Administrator (*SECADM) special authorities to use this command.

20

If you have *JOBCTL special authority, you do not need the specified authority to the object.

Service tools commands

This table lists the specific authorities required for the service tools commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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
CHGDSTPWD [1]			
CHGSSTSECA [2,3]			
CHGSSTUSR [2,3]			
CRTSSTUSR [2,3]			
DLTSSTUSR [2,3]			
DSPSSTSECA [4]			
DSPSSTUSR [4]			
STRSST [5] (Q)			

Command	Referenced object	Authority needed	
		For object	For library
1 You must be signed on with the QSECOFR user profile to use this command. 2 You must have *SECADM and *SERVICE special authorities. 3 The requesting service tools user ID must have the Service Tool user functional privilege "Service Tools Security". 4 You must have either *SECADM or *AUDIT special authority. 5 You must have *SERVICE special authority.			

Spelling aid dictionary commands

This table lists the specific authorities required for the 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)	Refer to the general rules.	*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

This table lists the specific authorities required for the 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

This table lists the specific authorities required for the spooled file commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, "Commands shipped with public authority *EXCLUDE," on page 355 shows which IBM-supplied user profiles are authorized to the command. The security officer can grant *USE authority to others.

Command	Referenced object	Output queue parameters			Special authority	Authority needed	
		DSPDTA	AUTCHK	OPRCTL		For object	For library
CHGSPLFA [1,2]	Output queue [3]		*DTAAUT			*READ, *DLT, *ADD	
			*OWNER			Owner [4]	
				*YES	*JOBCTL		
CHGSPLFA [1], if moving spooled file	Original output queue [3]		*DTAAUT			*READ, *ADD, *DLT	
			*OWNER			Owner [4]	
				*YES	*JOBCTL		
	Spooled file	*OWNER				Owner [6]	
	Target output queue[7]					*READ	*EXECUTE
				*YES	*JOBCTL		*EXECUTE
	Target device					*USE	
CPYSPLF [1]	Database file					Refer to the general rules for Display (DSP) or other operation using output file (OUTPUT (*OUTFILE))	Refer to the general rules for Display (DSP) or other operation using output file (OUTPUT (*OUTFILE))
	Spooled file	*OWNER				Owner [6]	
	Output queue [3]	*YES				*READ	
		*NO	*DTAAUT			*READ, *ADD, *DLT	
		*NO	*OWNER			Owner [4]	
		*YES or *NO		*YES	*JOBCTL		
DLTEXPSPLF (Q) [10]	Independent disk pool[9]					*USE	
DLTSPLF [1]	Output queue [3]		*DTAAUT			*READ, *ADD, *DLT	
			*OWNER			Owner [4]	
				*YES	*JOBCTL		

Command	Referenced object	Output queue parameters			Special authority	Authority needed	
		DSPDTA	AUTCHK	OPRCTL		For object	For library
DSPSPLF [1]	Output queue [3]	*YES				*READ	
		*NO	*DTAAUT			*READ, *ADD, *DLT	
		*NO	*OWNER			Owner [4]	
		*YES or *NO		*YES	*JOBCTL		
	Spooled file	*OWNER				Owner [6]	
HLDSPLF [1]	Output queue [3]		*DTAAUT			*READ, *ADD, *DLT	
			*OWNER			Owner [4]	
				*YES	*JOBCTL		
RCLSPLSTG (Q) [10]	Independent disk pool[9]					*USE	
RLSSPLF [1, 8]	Output queue [3]		*DTAAUT			*READ, *ADD, *DLT	
			*OWNER			Owner [4]	
				*YES	*JOBCTL		
SNDNETSPLF [1,5]	Output queue [3]	*YES				*READ	
		*NO	*DTAAUT			*READ, *ADD, *DLT	
		*NO	*OWNER			Owner [4]	
		*YES or *NO		*YES	*JOBCTL		
	Spooled file	*OWNER				Owner [6]	
SNDTCPSPLF [1,5]	Output queue [3]	*YES				*READ	
		*NO	*DTAAUT			*READ, *ADD, *DLT	
		*NO	*OWNER			Owner [4]	
		*YES or *NO		*YES	*JOBCTL		
	Spooled file	*OWNER				Owner [6]	
STRSPLRCL (Q)[9, 10]	Independent disk pool[9]					*USE	
WRKSPLF							

Command	Referenced object	Output queue parameters			Special authority	Authority needed	
		DSPDTA	AUTCHK	OPRCTL		For object	For library
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 You must be the owner of the spooled file. **7** If you have *SPLCTL special authority, you do not need authority to the target output queue but you must have *EXECUTE authority to its 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. **9** You must have *USE authority to all independent disk pools in an independent disk pool group. **10** You must have *SPLCTL special authority to run this command.							

Subsystem description commands

This table lists the specific authorities required for the subsystem description commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, "Commands shipped with public authority *EXCLUDE," on page 355 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[9]	*OBJOPR, *READ	*EXECUTE
ADDCMNE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
	Job description[9]	*OBJOPR, *READ	*EXECUTE
	User profile	*USE	
ADDJOBQE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE

Command	Referenced object	Authority needed	
		For object	For library
ADDPJE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
	User profile	*USE	
	Job description[9]	*OBJOPR, *READ	*EXECUTE
ADDRTGE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
ADDWSE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
	Job description[9]	*OBJOPR, *READ	*EXECUTE
CHGAJE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
	Job description[9]	*OBJOPR, *READ	*EXECUTE
CHGCMNE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
	Job description[9]	*OBJOPR, *READ	*EXECUTE
	User profile	*USE	
CHGJOBQE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
CHGPJE	Subsystem description	*OBJOPR, *OBJMGT, *READ	*EXECUTE
	User profile	*USE	
	Job description[9]	*OBJOPR, *READ	*EXECUTE
CHGRTGE	Subsystem description	*OJBOPR, *OBJMGT, *READ	*EXECUTE
CHGSBSD [5, 7]	Subsystem description	*OJBOPR, *OBJMGT, *READ	*EXECUTE
	signon display file[4]	*USE	*EXECUTE
CHGWSE	Subsystem description	*OJBOPR, *OBJMGT, *READ	*EXECUTE
	Job description[9]	*OBJOPR, *READ	*EXECUTE
CRTSBSD [5] (Q)	Subsystem description		*READ, *ADD
	signon display file [4]	*USE	*EXECUTE
	Auxiliary storage pool (ASP) device description [8]	*USE	
DLTSBSD	Subsystem description	*OBJEXIST, *USE	*EXECUTE
DSPSBSD	Subsystem description	*OBJOPR, *READ	*EXECUTE
ENDSBS [1]			
PRTSBSDAUT [6]			

Command	Referenced object	Authority needed	
		For object	For library
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	*EXECUTE
	Auxiliary storage pool (ASP) device description	*USE	
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 *SECADM or *ALLOBJ special authority to specify a specific library for the subsystem library.

6
You must have *ALLOBJ or *AUDIT special authority to use this command.

7
You must have *ALLOBJ and *SECADM special authorities to change the auxiliary storage pool (ASP) group name.

8
To specify an ASP device description that does not exist, you must have all object (*ALLOBJ) special authority.

9
To specify a job description that does not exist, you must have all object (*ALLOBJ) special authority.

System commands

This table lists the specific authorities required for the system commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. The Commands shipped with public authority *EXCLUDE topic 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
PWRDWNSYS[1]	Image catalog (if specified)	*USE	
RTVSYSINF (Q)[2]	Library	*READ, *ADD, *EXECUTE	
These commands do not require any object authorities:			
CHGSHRPOOL DSPSYSSTS ENDSYS[1] PRTSYSINF (Q)	RCLACTGRP[1] RCLRSC RETURN RTVGRPA	SIGNOFF UPDSYSINF (Q)[3] WRKSHRPOOL	WRKSYSSTS

1 You must have job control (*JOBCTL) special authority to use this command.

2 You must have *SAVSYS special authority to use this command.

3 You must have *SECADM, *ALLOBJ, *AUDIT, *JOBCTL, and *SAVSYS special authorities to use this command.

System reply list commands

This table lists the specific authorities required for the system reply list commands.

These commands do not require object authorities:			
ADDRPYLE (Q)	CHGRPYLE (Q)	RMVRPYLE (Q)	WRKRPYLE

System value commands

This table lists the specific authorities required for the system value commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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:			
CHGSYSVAL (Q) [1,2]	DSPSYSVAL[3]	RTVSYSVAL[3]	WRKSYSVAL [1,2, 3]

1

To change some system values, you must have *ALLOBJ, *ALLOBJ and *SECADM, *AUDIT, *IOSYSCFG, or *JOBCTL special authorities.

2

To use this command as shipped by IBM, you must be signed on as QPGMR, QSYSOPR, or QSRV, or have *ALLOBJ special authority.

3

To display or retrieve auditing-related system values, you must have either *AUDIT or *ALLOBJ special authority.

System/36 environment commands

This table lists the specific authorities required for the System/36 environment commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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
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)	Refer to the general rules.	*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)	Refer to the general rules.	*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

Command	Referenced object	Authority needed	
		For object	For library
CRTS36MNU	Menu: REPLACE(*NO)		*READ, *ADD, *CHANGE
	Menu: REPLACE(*YES)	Refer to the general rules.	*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)	Refer to the general rules.	*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 change attributes	*OBJMGT, *USE	*EXECUTE
	Program, to view attributes	*USE	*EXECUTE
EDTS36PRCA	File QS36PRC, to change attributes	*OBJMGT, *USE	*EXECUTE
	File QS36PRC, to view attributes	*USE	*EXECUTE

Command	Referenced object	Authority needed	
		For object	For library
EDTS36SRCA	Source file QS36SRC, to change attributes	*OBJMGT, *USE	*EXECUTE
	Source file QS36SRC, to view attributes	*USE	*EXECUTE
RSTS36F (Q)	From-file	*USE	*EXECUTE
	To-file	*ALL	Refer to the general rules.
	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	Refer to the general rules.
	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	Refer to the general rules.
	Device file or device description	*USE	*EXECUTE
SAVS36LIBM	From-file	*USE	*EXECUTE
	To-file, when it is a physical file	*ALL	Refer to the general rules.
	Device file or device description	*USE	*EXECUTE
WRKS36	S/36 configuration object QS36ENV	*READ	*EXECUTE
WRKS36PGMA	Program, to change attributes	*OBJMGT, *USE	*EXECUTE
	Program, to view attributes	*USE	*EXECUTE
WRKS36PRCA	File QS36PRC, to change attributes	*OBJMGT, *USE	*EXECUTE
	File QS36PRC, to view attributes	*USE	*EXECUTE
WRKS36SRCA	Source file QS36SRC, to change 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

This table lists the specific authorities required for the table commands.

Command	Referenced object	Authority needed	
		For object	For library
CRTTBL	Table		*READ, *ADD, *EXECUTE
	Source file	*USE	*EXECUTE
DLTTBL	Table	*OBJEXIST	*EXECUTE
WRKTBL [1]	Table	Any authority	*USE

1
To use an individual operation, you must have the authority required by the operation.

TCP/IP commands

This table lists the specific authorities required for the TCP/IP commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, "Commands shipped with public authority *EXCLUDE," on page 355 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
ADDTCPSVR[1]	Program to call	*EXECUTE	*EXECUTE
CHGTCPSVR[1]	Program to call	*EXECUTE	*EXECUTE
CPYTCPHT [6]	File objects		
CVTTCPCL (Q)	File objects	*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
RTVTCPINF[7] (Q)	Specified library	*READ, *ADD, *EXECUTE	
SETVTTBL	Table objects	*USE	*EXECUTE
SNDTCPSPLF [2]	Workstation customizing object	*USE	*EXECUTE
STRTCPFTP	Table objects	*USE	*EXECUTE
	File objects	*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
UPDTCPINF[8] (Q)	Specified library	*READ, *ADD, *EXECUTE	

These commands do not require any object authorities:

ADDCOMSNMP [1]
ADDNETTBLE [1]
ADDOSPFARA[1]
ADDOSPFLNK[1]
ADDOSPFIFC[1]
ADDOSPFRNG[1]
ADDPCLTBLE [1]
ADDRIPACP[1]
ADDRIPFLT[1]
ADDRIPIFC[1]
ADDRIPIGN[1]
ADDSDSTLE[11]
ADDSRVTBLE [1]
ADDTCPHTE [1]
ADDTCPIFC [1]
ADDTCPPORT [1]
ADDTCPRTE [1]
ADDUSRSMTP[9]
ADDUSRSNMP[1]
ARPING
CFGTCP
CFGTCPAPP
CFGTCPFTP [1]
CFGTCPLPD [1]
CFGRTG
CFGTCPSMTP
CFGTCPSNMP
CFGTCPTELN
CHGCOMSNMP [1]
CHGDHCPSVR[1]
CHGFTPA [1]
CHGLPDA [1]
CHGOSPFA[1]
CHGOSPFARA[1]
CHGOSPFIFC[1]
CHGOSPFLNK[1]
CHGOSPFRNG[1]
CHGRIPA[1]
CHGRIPFLT[1]
CHGRIPIFC[1]
CHGSDSTL[11]
CHGSMTPA [1]
CHGSNMPA [1]
CHGTCPA [1]
CHGTCPHTE[1]
CHGTCPIFC[1]
CHGTCPRTE [1]
CHGTELNA [1]
CHGUSRSMTP[9]
CHGUSRSNMP[1]
CHGVTMAP
CPYVPNCFGF[1]
CRTSDSTL[11]
DLTSDSTL[11]
DSPSDSTL[11]
DSPVTMAP
ENDTCP (Q)
ENDVPNCNN[1]
ENDTCPCNN
ENDTCPIFC (Q)
LODIPFTR[1]
MGRTCPHT [1]
NDPING
NETSTAT
PING
RMVCOMSNMP [1]
RMVNETTBLE [1]
RMVOSPFARA[1]
RMVOSPFIFC[1]
RMVOSPFLNK[1]
RMVOSPFRNG[1]
RMVPCLTBLE [1]
RMVRIPACP[1]
RMVRIPFLT[1]
RMVRIPIFC[1]
RMVRIPIGN[1]
RMVSDSTLE[11]
RMVSRVTBLE [1]
RMVTCPHTE[1]
RMVTCPIFC[1]
RMVTCPPORT [1]
RMVTCPRTE [1]
RMVTCPSVR [1]
RMVUSRSMTP[9]
RMVUSRSNMP[1]
RNMSDSTL[11]
RNMTCPHTE [1]
SETVTMAP
SNDARPRQS
SNDNGHSOL
SNDSMTPEMM[10]
STRIMPSMTP
STRTCP (Q)
STRTCPIFC (Q)
STRVPNCNN[1]
VFYTCPCNN
WRKNAMSMTP [3]
WRKNETTBLE[1]
WRKPCLTBLE[1]
WRKSDSTL[11]
WRKSMTPEMM[1]
WRKSMTPUSR[9]
WRKSRVTBLE[1]
WRKTCPSTS

These commands do not require any object authorities:
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.
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.
6 For the required authorities, refer to the description of the Display (DSP) or other operation using output file (OUTPUT(*OUTFILE)) section in the General rules for object authorities on commands topic.
7 You must have *SAVSYS special authority to use this command.
8 You must have *ALLOBJ, *SECADM, and *SAVSYS special authorities to use this command.
9 You must have *SECADM special authority to add, change, remove, or view entries for profiles different than the current user.
10 The current user profile must be enrolled in the e-mail directory that is set by the CHGSMTPA command and the DIRTYPE keyword. For a setting of *SDD for the DIRTYPE keyword the current user profile must be enrolled in the System Distribution Directory (SDD) and must also have an smtp name defined via the WRKNAMSMTP command. For a setting of *SMTP or *SMTPMSF the current user profile must be enrolled via ADDUSRSMTP.
11 You must have *SECADM special authority to change, delete, rename, add entries to, or remove entries from a distribution list which you do not own.

Time zone description commands

This table lists the specific authorities required for the time zone description commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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
CHGTIMZON	Time zone description	*CHANGE	*EXECUTE
CRTTIMZON	Time zone description		*READ, *ADD
DLTTIMZON[1]	Time zone description	*OBJEXIST	*EXECUTE
WRKTIMZON[2]	Time zone description	*USE	*USE

Command	Referenced object	Authority needed	
		For object	For library

1
The time zone description specified in the QTIMZON system value cannot be deleted.

2
If a message is used to specify the abbreviated and full names of the time zone description, you must have *USE authority to the message file and *EXECUTE authority to the message file's library in order to see the abbreviated and full names.

User index, user queue, and user space commands

This table lists the specific authorities required for the user index, user queue, and user space commands.

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-defined file system commands

This table lists the specific authorities required for the user-defined file system commands.

Command	Referenced object	Object type	File system	Authority needed for object
ADDMFS [1,2,3]	dir_to_be_mounted_over	*DIR	"root" (/)	*W
	Path Prefix	Refer to the general rules.		
CRTUDFS [1,2,6,7] (Q)	/dev/QASP*xx* or /dev/IASPname	*DIR	"root" (/)	*RWX
DLTUDFS [1,2,4,5,8,9,10] (Q)	/dev/QASP*xx* or /dev/IASPname	*DIR	"root" (/)	*RWX
	any integrated file system object in the UDFS		"root" (/)	*OBJEXIST
	Any non-empty directory object	*DIR	"root" (/)	*WX
DSPUDFS	some_dirsxx	*DIR	"root" (/)	*RX
MOUNT [1,2,3]	dir_to_be_ mounted_over	*DIR	"root" (/)	*W
	Path Prefix	Refer to the general rules.		
RMVMFS [1]				
UNMOUNT [1]				

Command	Referenced object	Object type	File system	Authority needed for object
1 To use this command, you must have *IOSYSCFG special authority. **2** There are two directory naming conventions depending on the location of the user-defined file system (UDFS). Use one of the following conventions: • - /dev/QASPxx where *xx* is 01 for the system asp or 02-32 for the basic user asps. • - /dev/IASPname where *IASPname* is the name of the independent ASP. 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 integrated file system directory that can be mounted over. **4** A UDFS can contain an entire subtree of objects, so when you delete a UDFS, you delete objects of all types that can be stored in the user-defined file system. **5** When using the DLTUDFS commands, you must have *OBJEXIST authority on every object in the UDFS or no objects are deleted.				
6 You must have all object (*ALLOBJ) and security administrator (*SECADM) special authorities to specify a value for the Scanning option for objects (CRTOBJSCAN) parameter other than *PARENT. **7** The audit (*AUDIT) special authority is required when specifying a value other than *SYSVAL on the Auditing value for objects (CRTOBJAUD) parameter. **8** You must have write (*W) and execute (*X) authority to all of the non-empty directory objects in the UDFS. **9** If any non-empty directory object in the UDFS has the "restricted rename and unlink" attribute set to Yes (this attribute is equivalent to the S_ISVTX mode bit), then one or more of the following conditions must be true: • You must be the owner of all the objects contained in the directory. • You must be the owner of the directory. • You must have all object (*ALLOBJ) special authority. **10** The UDFS cannot be deleted if it contains an object with the *read only* attribute set to *yes* or if it contains an object that is checked out.				

User profile commands

This table lists the specific authorities required for the user profile commands.

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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)			
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			

Command	Referenced object	Authority needed	
		For object	For library
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
CVTUSRCERT[3, 14]			
DLTUSRPRF [3,9]	User profile	*OBJEXIST, *USE	*EXECUTE
	Message queue [5]	*OBJEXIST, *USE, *DLT	*EXECUTE
DMPUSRPRF [22](Q)	User profile		
DSPACTPRFL [14](Q)			
DSPACTSCD [14](Q)			
DSPAUTUSR [6]	User profile	*READ	
DSPEXPSCD [14](Q)			
DSPPGMADP	User profile	*OBJMGT	
	Output file	Refer to the general rules.	Refer to the general rules.
DSPSSTUSR [23]			
DSPUSRPRF[19]	User profile	*READ	*EXECUTE
	Output file	Refer to the general rules.	Refer to the general rules.
DSPUSRPRTI	User profile	*USE	
GRTUSRAUT [7]	Referenced user profile	*READ	
	Objects you are granting authority to	*OBJMGT	*EXECUTE
PRTPRFINT [14](Q)			
PRTUSRPRF [18]			
RSTAUT (Q) [8]			
RSTUSRPRF (Q) [8,10, 16]			
RTVUSRPRF[20]	User profile	*READ	
RTVUSRPRTI	User profile	*USE	

Command	Referenced object	Authority needed	
		For object	For library
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.

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.

10

You must have *ALLOBJ special authority to specify a value other than *NONE for the Allow object differences (ALWOBJDIF) parameter.

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.

Command	Referenced object	Authority needed	
		For object	For library
16 You must have *ALLOBJ and *SECADM special authorities to specify SECDTA(*PWDGRP), USRPRF(*ALL) or OMITUSRPRF. **17** When you perform a CRTUSRPRF, you cannot 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. **18** You must have *ALLOBJ or *AUDIT special authority to use this command. **19** You must have either *ALLOBJ or *AUDIT special authority to display the current object auditing value and action auditing value displayed. Otherwise, the value *NOTAVL is displayed to indicate that the values are unavailable for display. **20** You must have either *ALLOBJ or *AUDIT special authority to retrieve the current OBJAUD and AUDLVL values. Otherwise, the value *NOTAVL is returned to indicate that the values are unavailable for retrieval.			
21 To use this command, you must have service (*SERVICE) special authority, or be authorized to the Service Dump function of IBM i through the support of the IBM Navigator for i Application Administration. The Change Function Usage (**CHGFCNUSG**) command with a function ID of QIBM_SERVICE_DUMP can also be used to change the list of users that are allowed to perform dump operations. **22** To use this command, you must have *SERVICE special authority or have the authorization to the QIBM_SERVICE_DUMP function usage list. **23** You must have either security administrator (*SECADM) or audit (*AUDIT) special authority to use this command.			

Validation list commands

This table lists the specific authorities required for the validation list commands.

Command	Referenced object	Authority needed	
		For object	For library
CRTVLDL	Validation list		*ADD, *READ
DLTVLDL	Validation list	*OBJEXIST	*EXECUTE

Workload capping group commands

This table lists the specific authorities required for the workload capping group commands

Commands identified by (Q) are shipped with public authority *EXCLUDE. Appendix C, “Commands shipped with public authority *EXCLUDE,” on page 355 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.

Command	Referenced object	Authority needed	
		For object	For library
ADDWLCGRP [1] (Q)			
ADDWLCPRDE (Q)			
CHGWLCGRP [1] (Q)			
DSPWLCGRP [1] (Q)			
RMVWLCGRP [1] (Q)			
RMVWLCPRDE (Q)			

1
You must have *JOBCTL special authority to use this command.

Workstation customization commands

This table lists the specific authorities required for the workstation customization 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

This table lists the specific authorities required for the writer commands.

Command	Referenced object	Output queue parameters		Special authority	Authority needed	
		AUTCHK	OPRCTL		For object	For library
CHGWTR [2, 4]	Current output queue[1]	*DTAAUT			*READ, *ADD, *DLT	
		*OWNER			Owner [3]	
			*YES	*JOBCTL		
	New output queue	*DTAAUT			*READ, *ADD, *DLT	*EXECUTE
		*OWNER			Owner	*EXECUTE
			*YES	*JOBCTL		*EXECUTE

Command	Referenced object	Output queue parameters		Special authority	Authority needed	
		AUTCHK	OPRCTL		For object	For library
ENDWTR [1]	Output queue	*DTAAUT			*READ, *ADD, *DLT	
		*OWNER			Owner [3]	
			*YES	*JOBCTL		
HLDWTR [1]	Output queue	*DTAAUT			*READ, *ADD, *DLT	
		*OWNER			Owner [3]	
			*YES	*JOBCTL		
RLSWTR [1]	Output queue	*DTAAUT			*READ, *ADD, *DLT	
		*OWNER			Owner [3]	
			*YES	*JOBCTL		
STRDKTWTR [1]	Output queue	*DTAAUT			*READ, *ADD, *DLT	*EXECUTE
		*OWNER			Owner [3]	*EXECUTE
			*YES	*JOBCTL		*EXECUTE
	Message queue				*OBJOPR, *ADD	*EXECUTE
	Device description				*OBJOPR, *READ	
STRPRTWTR [1]	Output queue	*DTAAUT			*READ, *ADD, *DLT	*EXECUTE
		*OWNER			Owner [3]	*EXECUTE
			*YES	*JOBCTL		*EXECUTE
	Message queue				*OBJOPR, *ADD	*EXECUTE
	Workstation customization object				*USE	*EXECUTE
	User-driver program				*OBJOPR *EXECUTE	*EXECUTE
	User-data transform program				*OBJOPR *EXECUTE	*EXECUTE
	User separator program				*OBJOPR *EXECUTE	*EXECUTE
	Device Description				*OBJOPR, *READ	

Command	Referenced object	Output queue parameters		Special authority	Authority needed	
		AUTCHK	OPRCTL		For object	For library
STRRMTWTR [1]	Output queue	*DTAAUT			*READ, *ADD, *DLT	*EXECUTE
		*OWNER			Owner [3]	*EXECUTE
			*YES	*JOBCTL		*EXECUTE
	Message queue				*OBJOPR, *ADD	*EXECUTE
	Workstation customization object				*USE	*EXECUTE
	User-driver program				*OBJOPR *EXECUTE	*EXECUTE
	User-data transform program				*OBJOPR *EXECUTE	*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 topic collection 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, including 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 296 describes how to set up auditing for objects.

Please also refer to section "Relationship of object Change Date/Time to audit records" on page 307.

Operations shown in the tables in uppercase, such as CPYF, refer to CL commands, unless they are labeled as an application programming interface (API).

Related concepts

Using the security audit journal
The security audit journal is the primary source of auditing information about the system. This section describes how to plan, set up, and manage security auditing, what information is recorded, and how to view that information.

Operations common to all object types

This list describes the operations that you can perform against all object types, and whether those operations are audited.

- Read operation

  CRTDUPOBJ
  Create Duplicate Object (if *ALL is specified for *"from-object"*).

  DMPOBJ
  Dump Object

  DMPSYSOBJ
  Dump System Object

  QSRSAVO
  Save Object API

  QsrSave
  Save Object in Directory API

  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

CHGJRNOBJ
Change Journaled Object

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. If the object is being created into QTEMP library then a CO entry is not written.

DLTxxxxxx or DLTOBJ
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. If the object is being deleted from QTEMP library then a DO entry is not written.

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

QLICOBJD
Change Object Description API

QLIRNMO
Rename Object API

QjoEndJournal
End Journaling

QjoStartJournal
Start Journaling

QSRRSTO
Restore Object API

QsrRestore
Restore Object in Directory API

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 [1]
Prompt override program for a change command (if one exists)

CHKOBJ
Check Object

ALCOBJ
Allocate Object

CPROBJ
Compress Object

DCPOBJ
Decompress Object

DLCOBJ
Deallocate Object

DSPOBJD
Display Object Description

DSPOBJAUT
Display Object Authority

EDTOBJAUT
Edit Object Authority

Note: If object authority is changed and action auditing includes *SECURITY, or the object is being audited, an audit record is written.

[1] A prompt override program displays the current values when prompting is requested for a command. For example, if you type CHGURSPRF USERA and press F4 (prompt), the Change User Profile display shows the current values for the USERA user profile.

QSYCUSRA
Check User's Authority to an Object API

QSYLUSRA
List Users Authorized to an Object API. An audit record is not written for the object whose authority is being listed. An audit record is written for the user space used to contain information.

QSYRUSRA
Retrieve User's Authority to Object API

RCLTMPSTG
Reclaim Temporary Storage

RMVDFRID
Remove Defer ID

RSTDFROBJ
Restore Deferred Object

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

This list describes the operations that you can perform against the Access Path Recovery Times object, and whether those operations are audited.

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)

This list describes the operations that you can perform against Alert Table (*ALRTBL), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Authorization List (*AUTL), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Authority Holder (*AUTHLR), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Binding Directory (*BNDDIR), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Configuration List (*CFGL), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Special Files (*CHRSF), and whether those operations are audited.

See Operations for Stream File (*STMF) for *CHRSF auditing.

Operations for Chart Format (*CHTFMT)

This list describes the operations that you can perform against Chart Format (*CHTFMT), and whether those operations are audited.

- Read operation

Display
DSPCHT command or option F10 from the BGU menu

Print/Plot
DSPCHT command or option F15 from the BGU menu

Save/Create
Save or create graphics data file (GDF) using CRTGDF command or option F13 from the BGU menu

- Change operation

None

- Operations that are not audited

None

Operations for C Locale Description (*CLD)

This list describes the operations that you can perform against C Locale Description (*CLD), and whether those operations are audited.

- 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 Change Request Description (*CRQD)

This list describes the operations that you can perform against Change Request Description (*CRQD), and whether those operations are audited.

- 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 Description Activity

- Operations that are not audited

  WRKCRQD
  Work with Change Request Descriptions

Operations for Class (*CLS)

This list describes the operations that you can perform against Class (*CLS), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Command (*CMD), and whether those operations are audited.

- 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. The use of these commands is not audited.

CALL [1]
CALLPRC
CHGVAR
COPYRIGHT
DCL
DCLF
DO
ELSE
ENDDO
	ENDPGM
	ENDRCV
	GOTO
	IF
	MONMSG
	PGM
		RCVF
		RETURN
		SNDF
		SNDRCVF
		TFRCTL
		WAIT

1 CALL is audited if it is run interactively. It is not audited if it is run within a CL program.

Operations for Connection List (*CNNL)

This list describes the operations that you can perform against Connection List (*CNNL), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Class-of-Service Description (*COSD), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Communications Side Information (*CSI), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Cross System Product Map (*CSPMAP), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Cross System Product Table (*CSPTBL), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Controller Description (*CTLD), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Device Description (*DEVD), and whether those operations are audited.

- Read operation

  Acquire
  First acquisition 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)

This list describes the operations that you can perform against Directory (*DIR) objects, and whether those operations are audited.

- Read/search operations

access, accessx, QlgAccess, QlgAccessx
Determine file accessibility

CHGATR
Change Attribute

CPY
Copy Object

DSPCURDIR
Display Current Directory

DSPLNK
Display Object Links

faccessx
Determine file accessibility for a class of users by descriptor

getcwd, qlgGetcwd
Get Path Name of Current Directory API

Qp0lGetAttr, QlgGetAttr
Get attributes APIs

Qp0lGetPathFromFileID, QlgGetPathFromFileID
Get Path From File Identifier 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 Object

WRKLNK
Work with Links

- Change operation

CHGATR
Change Attributes

CHGAUD
Change Auditing Value

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 Object

CRTDIR
Make Directory

fchmod
Change File Authorizations by Descriptor API

fchown
Change Owner and Group of File by Descriptor API

mkdir, QlgMkdir
Make Directory API

MOV
Move Object

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 Object

RST
Restore Object

utime, QlgUtime
Set File Access and Modifcation Times API

WRKAUT
Work with Authority

WRKLNK
Work with Object 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 Server

This list describes the operations that you can perform against Directory Server, and whether those operations are audited.

Note: Directory Server 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 can 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)

This list describes the operations that you can perform against document library objects (*DOC or *FLR), and whether those operations are audited.

- 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 [2]
Copy Document

Note: A change entry is written if the target document already exists.

CRTFLR
Create Folder

CVTTOFLR [2]
Convert to Folder

[2] A change entry is written for both the document and the folder if the target of the operation is in a folder.

DLTDLO [2]
Delete DLO

DLTSHF
Delete Bookshelf

DTLDOCL [2]
Delete Document List

DLTDST [2]
Delete Distribution

EDTDLOAUT
Edit DLO Authority

EDTDOC
Edit Document

FILDOC [2]
File Document

GRTACCAUT
Grant Access Code Authority

GRTUSRPMN
Grant User Permission

MOVDOC [2]
Move Document

MRGDOC [2]
Merge Document

PAGDOC
Paginate Document

QHFCHGAT
Change Directory Entry Attributes API

QHFSETSZ
Set Stream File Size API

QHFWRTSF
Write Stream File API

QRYDOCLIB [2]
Query Document Library

Note: A change entry is written if an existing document resulting from a search is replaced.

RCVDST [2]
Receive Distribution

RGZDLO
Reorganize DLO

RMVACC
Remove access code, for any DLO to which the access code is attached

RMVDLOAUT
Remove DLO authority

RNMDLO [2]
Rename DLO

RPLDOC
Replace Document

RSTDLO [2]
Restore DLO

RSTSHF
Restore Bookshelf

RTVDOC
Retrieve Document (check out)

RVKACCAUT
Revoke Access Code Authority

RVKUSRPMN
Revoke User Permission

SAVDLO [2]
Save DLO

- Operations that are not audited

ADDACC
Add Access Code

DSPACC
Display Access Code

DSPUSRPMN
Display User Permission

QHFCHGFP
Change File Pointer API

QHFCLODR
Close Directory API

QHFCLOSF
Close Stream File API

QHFFRCSF
Force Buffered Data API

QHFLULSF
Lock/Unlock Stream File Range API

QHFRTVAT
Retrieve Directory Entry Attributes API

RCLDLO
Reclaim DLO (*ALL or *INT)

WRKDOCLIB
Work with Document Library

WRKDOCPRTQ
Work with Document Print Queue

Operations for Data Area (*DTAARA)

This list describes the operations that you can perform against Data Area (*DTAARA), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Interactive Data Definition Utility (*DTADCT), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Data Queue (*DTAQ), and whether those operations are audited.

- 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

QMHQCDQ
Change Data Queue API

- Operations that are not audited

WRKDTAQ
Work with Data Queue

QMHQRDQD
Retrieve Data Queue Description API

Operations for Edit Description (*EDTD)

This list describes the operations that you can perform against Edit Description (*EDTD), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Exit Registration (*EXITRG), and whether those operations are audited.

- 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
Unregister Exit Point API

QusDeregisterExitPoint
Unregister 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)

This list describes the operations that you can perform against Forms Control Table (*FCT), and whether those operations are audited.

- No Read or Change operations are audited for the *FCT object type.

Operations for File (*FILE)

This list describes the operations that you can perform against File (*FILE), and whether those operations are audited.

- 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

APYJRNCHGX
Apply Journal Changes Extend

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

CHGPFTRG
Change Physical File Trigger

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)

This list describes the operations that you can perform against first-in first-out (*FIFO) objects, and whether those operations are audited.

See Operations for Stream File (*STMF) for the *FIFO auditing.

Operations for Folder (*FLR)

This list describes the operations that you can perform against folder (*FLR) objects, and whether those operations are audited.

See operations for "Operations for Document Library Object (*DOC or *FLR)" on page 581

Operations for Font Resource (*FNTRSC)

This list describes the operations that you can perform against Font Resource (*FNTRSC), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Form Definition (*FORMDF), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Filter Object (*FTR), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Graphics Symbols Set (*GSS), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Double-byte Character Set Dictionary (*IGCDCT), and whether those operations are audited.

- Read operation

DSPIGCDCT
Display IGC Dictionary

- Change operation

EDTIGCDCT
Edit IGC Dictionary

Operations for Double-byte Character Set Sort (*IGCSRT)

This list describes the operations that you can perform against Double-byte Character Set Sort (*IGCSRT), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Double-byte Character Set Table (*IGCTBL), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Job Description (*JOBD), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Job Queue (*JOBQ), and whether those operations are audited.

- Read operation

  None

- Change operation

  Entry
  When an entry is placed on or removed from the queue

  CHGJOBQ
  Change Job Queue

  CLRJOBQ
  Clear Job Queue

  HLDJOBQ
  Hold Job Queue

  RLSJOBQ
  Release Job Queue

- Operations that are not audited

  ADDJOBQE **"Subsystem descriptions" on page 206**
  Add Job Queue Entry

  CHGJOB
  Change Job from one JOBQ to another JOBQ

  CHGJOBQE **"Subsystem descriptions" on page 206**
  Change Job Queue Entry

  QSPRJOBQ
  Retrieve job queue information

  RMVJOBQE **"Subsystem descriptions" on page 206**
  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

  WRKJOBQD
  Work with Job Queue Description

Operations for Job Scheduler Object (*JOBSCD)

This list describes the operations that you can perform against Job Scheduler Object (*JOBSCD), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Journal (*JRN), and whether those operations are audited.

- 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

[3] An audit record is written if object auditing is specified for the subsystem description (*SBSD).

APYJRNCHG
Apply Journaled Changes

APYJRNCHGX
Apply Journal Changes Extend

CHGJRN
Change Journal

CHGRMTJRN
Change Remote Journal

ENDJRNxxx
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)

STRJRNxxx
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)

This list describes the operations that you can perform against Journal Receiver (*JRNRCV), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Library (*LIB), and whether those operations are audited.

- Read operation

  DSPLIB
  Display Library (when library is not empty. If library is empty, no audit is performed.)

  Locate
  When a library is accessed to find an object

  Note:

  1. Several audit entries might 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 OBJ(AR/WRK*) OBJTYPE(*FILE)
     ```

     If a library named "AR" does not have any file names beginning with "WRK", no audit record is written for that library.

  Library list
  Adding library to a library list

- Change operation

  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)

This list describes the operations that you can perform against Line Description (*LIND), and whether those operations are audited.

- Read operation

SAVCFG
Save Configuration

RUNLPDA
Run LPDA-2 operational commands

VFYCMN
Link test

VFYLNKLPDA
LPDA-2 link test

- Change operation

CHGLINxxx
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

This list describes the operations that you can perform against Mail Services, and whether those operations are audited.

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)

This list describes the operations that you can perform against Menu (*MENU), and whether those operations are audited.

- Read operation

  Display
  Displaying a menu through the GO MENU command or UIM dialog box 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)

This list describes the operations that you can perform against Mode Description (*MODD), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Module Object (*MODULE), and whether those operations are audited.

- 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

  RTVCLSRC
  An audit entry for each module object used during a RTVCLSRC

  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

Module Conversion
Machine-initiated conversion for compatibility with the current machine

RTVBNDSRC
Retrieve Binder Source

WRKMOD
Work with Module

Operations for Message File (*MSGF)

This list describes the operations that you can perform against Message File (*MSGF), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Message Queue (*MSGQ), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Node Group (*NODGRP), and whether those operations are audited.

- Read operation

  DSPNODGRP
  Display Node Group

- Change operation

  CHGNODGRPA
  Change Node Group

Operations for Node List (*NODL)

This list describes the operations that you can perform against Node List (*NODL), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against NetBIOS Description (*NTBD), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Network Interface (*NWID), and whether those operations are audited.

- 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

  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)

This list describes the operations that you can perform against Network Server Description (*NWSD), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Output Queue (*OUTQ), and whether those operations are audited.

- 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 [4]
  Change Spooled File Attributes, if moved to a different output queue and either output queue is audited

  CLROUTQ
  Clear Output Queue

  DLTSPLF [4]
  Delete Spooled File

  HLDOUTQ
  Hold Output Queue

  RLSOUTQ
  Release Output Queue

- Operations that are not audited

  CHGSPLFA [4]
  Change Spooled File Attributes

  CPYSPLF [4]
  Copy Spooled File

  Create [4]
  Create a spooled file

  DSPSPLF [4]
  Display Spooled File

  HLDSPLF [4]
  Hold Spooled File

  QSPROUTQ
  Retrieve output queue information

  RLSSPLF [4]
  Release Spooled File

  SNDNETSPLF [4]
  Send Network Spooled File

  WRKOUTQ
  Work with Output Queue

  WRKOUTQD
  Work with Output Queue Description

  WRKSPLF
  Work with Spooled File

WRKSPLFA
Work with Spooled File Attributes

Operations for Overlay (*OVL)

This list describes the operations that you can perform against Overlay (*OVL), and whether those operations are audited.

- Read operation

  Print
  Printing a spooled file that refers to the overlay

- Change operation

  None

- Operations that are not audited

  WRKOVL
  Work with overlay

  Print
  Referring to the overlay when creating a spooled file

Operations for Page Definition (*PAGDFN)

This list describes the operations that you can perform against Page Definition (*PAGDFN), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Page Segment (*PAGSEG), and whether those operations are audited.

- Read operation

  Print
  Printing a spooled file that refers to the page segment

- Change operation

  None

- Operations that are not audited

[4] This is also audited if action auditing (QAUDLVL system value or AUDLVL user profile value) includes *SPLFDTA.

WRKPAGSEG
Work with Page Segment

Print
Referring to the page segment when creating a spooled file

Operations for Print Descriptor Group (*PDG)

This list describes the operations that you can perform against Print Descriptor Group (*PDG), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Program (*PGM), and whether those operations are audited.

- 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

Program Conversion
Machine-initiated conversion for compatibility with the current machine

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)

This list describes the operations that you can perform against Panel Group (*PNLGRP), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Product Availability (*PRDAVL), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Product Definition (*PRDDFN), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Product Load (*PRDLOD), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Query Manager Form (*QMFORM), and whether those operations are audited.

- 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 Db2 Query Manager and SQL Development Kit for IBM i.

- 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)

This list describes the operations that you can perform against Query Manager Query (*QMQRY), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Query Definition (*QRYDFN), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Reference Code Translate Table (*RCT), and whether those operations are audited.

- Read operation

None

- Change operation

None

- Operations that are not audited

None

Operations for Reply List

This list describes the operations that you can perform against Reply List, and whether those operations are audited.

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)

This list describes the operations that you can perform against Subsystem Description (*SBSD), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Information Search Index (*SCHIDX), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Local Socket (*SOCKET), and whether those operations are audited.

- 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

Note: Close is not audited, but if there were a failure or modification in a close scan_related exit program, then an audit record is cut.

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)

This list describes the operations that you can perform against Spelling Aid Dictionary (*SPADCT), and whether those operations are audited.

- Read operation

  Verify
  Spell verify function

  Aid
  Spell aid function

  Hyphenation
  Hyphenation function

  Dehyphenation
  Dehyphenation function

  Synonyms
  Synonym function

  Base
  Use dictionary as base when creating another dictionary

  Verify
  Use 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

This list describes the operations that you can perform against Spooled Files, and whether those operations are audited.

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 with CHGSPLFA:

  - COPIES

- DEV
- FORMTYPE
- RESTART
- PAGERANGE
- OUTQ
- DRAWER
- PAGDFN
- FORMDF
- USRDFNOPT
- USRDFNOBJ
- USRDFNDTA
- EXPDATE
- SAVE

Changing any other spooled file attributes with CHGSPLFA:

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 operations:

- 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
- Deleting of spooled files that have expired using the DLTEXPSPLF command
- Deleting of spooled files through the operational assist cleanup function

Hold
Holding a spooled file by any of the following operations:

- 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

Restore
Restoring a spooled file

Save
Saving a spooled file

Operations for SQL Package (*SQLPKG)

This list describes the operations that you can perform against SQL Package (*SQLPKG), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Service Program (*SRVPGM), and whether those operations are audited.

- 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

  RTVCLSRC
  An audit entry for each service program used during a RTVCLSRC command

  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

Service Program Conversion
Machine-initiated conversion for compatibility with the current machine

QBNLSPGM
List Service Program Information API

QBNRSPGM
Retrieve Service Program Information API

WRKSRVPGM
Work with Service Program

Operations for Session Description (*SSND)

This list describes the operations that you can perform against Session Description (*SSND), and whether those operations are audited.

No Read or Change operations are audited for the *SSND object type.

Operations for Server Storage Space (*SVRSTG)

This list describes the operations that you can perform against Server Storage Space (*SVRSTG), and whether those operations are audited.

No Read or Change operations are audited for the *SVRSTG object type.

Operations for Stream File (*STMF)

This list describes the operations that you can perform against Stream File (*STMF) objects, and whether those operations are audited.

- Read operation

  CPY
  Copy Object

  DSPLNK
  Display Object Links

  givedescriptor
  Give File Access API

  MOV
  Move Object

  open, open64, QlgOpen, QlgOpen64, Qp0lOpen
  Open File APIs

  SAV
  Save Object

  WRKLNK
  Work with Object Links

- Change operation

  ADDLNK
  Add Link

  CHGAUD
  Change Auditing

  CHGAUT
  Change Authority

  CHGOWN
  Change Owner

CHGPGP
Change Primary Group

CHKIN
Check In Object

CHKOUT
Check Out Object

chmod, QlgChmod
Change File Authorizations APIs

chown, QlgChown
Change Owner and Group APIs

CPY
Copy Object

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 Object

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 Object

RST
Restore Object

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

fclear, fclear64
Clear a file

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)

This list describes the operations that you can perform against symbolic link (*SYMLNK) objects, and whether those operations are audited.

- Read operation

  CPY
  Copy Object

  DSPLNK
  Display Object Links

  MOV
  Move Object

  readlink
  Read Value of Symbolic Link API

  SAV
  Save Object

  WRKLNK
  Work with Object Links

- Change operation

  CHGOWN
  Change Owner

  CHGPGP
  Change Primary Group

  CPY
  Copy Object

  MOV
  Move Object

  Qp0lRenameKeep, QlgRenameKeep
  Rename File or Directory, Keep New APIs

  Qp0lRenameUnlink, QlgRenameUnlink
  Rename File or Directory, Unlink New APIs

  RMVLNK
  Remove Link

  RNM
  Rename Object

  RST
  Restore Object

  symlink, QlgSymlink
  Make Symbolic Link APIs

  unlink, QlgUnlink
  Remove Link to File APIs

  WRKLNK
  Work with Object Links

- Operations that are not audited

  lstat, lstat64, QlgLstat, QlgLstat64
  Link Status APIs

Operations for S/36 Machine Description (*S36)

This list describes the operations that you can perform against S/36 Machine Description (*S36), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Table (*TBL), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against User Index (*USRIDX), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against User Profile (*USRPRF), and whether those operations are audited.

- Read operation

  RCLOBJOWN
  Reclaim Objects by Owner

- 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)

This list describes the operations that you can perform against User Queue (*USRQ), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against User Space (*USRSPC), and whether those operations are audited.

- 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)

This list describes the operations that you can perform against Validation List (*VLDL), and whether those operations are audited.

- 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 for Workstation Customizing Object (*WSCST)

This list describes the operations that you can perform against Workstation Customizing Object (*WSCST), and whether those operations are audited.

- 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 section 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 journal entry layouts described in this appendix are similar to how one can define a physical file using DDS. For instance, a Binary (4) is defined to hold from 1 to 4 digits information with the storage requirement of two bytes, while a Binary (5) holds from 1 to 5 digits information with the storage requirement of 4 bytes. Languages such as RPG use and enforce these definitions. 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 Journal management topic.

"Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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.

"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632 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.

Note: TYPE2 and *TYPE4 output formats are no longer updated; therefore, it is recommended that you stop using *TYPE2 and *TYPE4 formats and use only *TYPE5 formats.

"Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630 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.

"AD (Auditing Change) journal entries" on page 637 through "ZR (Read of Object) journal entries" on page 886 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 305 provides examples of using the model database outfiles. See also the Journal management topic.

Note: In these journal entries tables, you might see a blank column under the offset, JE or J4, column. It means there is no model outfile for that audit journal type.

Related concepts

Using the security audit journal

The security audit journal is the primary source of auditing information about the system. This section describes how to plan, set up, and manage security auditing, what information is recorded, and how to view that information.

Related information

Journal management

Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)

This table lists all possible values for the fields that are common to all entry types when OUTFILFMT(*TYPE5) is specified on the DSPJRN command.

Table 158. 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 "Audit Journal (QAUDJRN) entry types" on page 635 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.[2]
65	User Name	Char(10)	The user profile name associated with the job.[1,2]
75	Job Number	Zoned(6,0)	The job number.[2]
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 conditions 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].

Table 158. Standard heading fields for audit journal entries. QJORDJE5 Record Format (*TYPE5) *(continued)*			
Offset	**Field**	**Format**	**Description**
197	System Name	Char(8)	The name of the system.
205	Journal identifier	Char(10)	Used for journaled objects. Not used for audit journal entries.
215	Referential Constraint	Char(1)	Used for journaled objects. Not used for audit journal entries.
216	Trigger	Char(1)	Used for journaled objects. Not used for audit journal entries.
217	Incomplete Data	Char(1)	Used for journaled objects. Not used for audit journal entries.
218	Ignored by APY/ RMVJRNCHG	Char(1)	Used for journaled objects. Not used for audit journal entries.
219	Minimized ESD	Char(1)	Used for journaled objects. Not used for audit journal entries.
220	Object indicator	Char(1)	Used for journaled objects. 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 that holds 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 that holds 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 journaled objects. Not used for audit journal entries.
396	Transaction ID	Char(140)	Used for journaled objects. Not used for audit journal entries.
536	Reserved	Char(20)	Used for journaled objects. Not used for audit journal entries.
556	Null value indicators[3]	Char(52)	Used for journaled objects. Not used for audit journal entries.
608	Entry specific data length	Binary(4)	Length of the entry specific data.

Table 158. Standard heading fields for audit journal entries. QJORDJE5 Record Format (*TYPE5) *(continued)*			
Offset	**Field**	**Format**	**Description**
1			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 exchange user profiles, the User profile name field contains the name of the new (swapped) user profile.
2			If the system job is running in a task rather than a process, the name of job and user name fields that begin at offset 55 contain up to a 16 character name for the LIC task. The remaining characters of the user name field that start at offset 71 are left blank. The job number field that begins at offset 75 is set to zeros.
3			This is a variable length field. The first 2 bytes contain the length of the null value indicators.

Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)

This table lists all possible values for the fields that are common to all entry types when OUTFILFMT(*TYPE4) is specified on the DSPJRN command.

Table 159. 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 "Audit Journal (QAUDJRN) entry types" on page 635 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.[2]
55	User Name	Char(10)	The user profile name associated with the job.[1,2]
65	Job Number	Zoned(6,0)	The job number.[2]
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.

Table 159. Standard heading fields for audit journal entries. QJORDJE4 Record Format (*TYPE4) *(continued)*			
Offset	**Field**	**Format**	**Description**
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	Journal Identifier	Char(10)	Used for journaled objects. Not used for audit journal entries.
160	Referential Constraint	Char(1)	Used for journaled objects. Not used for audit journal entries.
161	Trigger	Char(1)	Used for journaled objects. Not used for audit journal entries.
162	(Reserved Area)	Char(8)	
170	Null Value Indicators[3]	Char(52)	Used for journaled objects. Not used for audit journal entries.
222	Entry Specific Data Length	Binary (4)	Length of the entry specific data.

1

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 exchange user profiles, the User profile name field contains the name of the new (swapped) user profile.

2

If the system job is running in a task rather than a process, the name of job and user name fields that begin at offset 45 contain up to a 16 character name for the LIC task. The remaining characters of the user name field that start at offset 61 are left blank. The job number field that begins at offset 65 is set to zeros.

3

This is a variable length field. The first 2 bytes contain the length of the null value indicators.

Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)

This table lists all possible values for the fields that are common to all entry types when OUTFILFMT(*TYPE2) is specified on the DSPJRN command.

Table 160. 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.

Table 160. Standard heading fields for audit journal entries. QJORDJE2 Record Format (*TYPE2) *(continued)*

Offset	Field	Format	Description
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 "Audit Journal (QAUDJRN) entry types" on page 635 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 exchange user profiles, the *User profile name* field contains the name of the new (swapped) user profile.

Audit Journal (QAUDJRN) entry types

This table introduces all available entry types for the audit journal.

Table 161. Audit Journal (QAUDJRN) entry types

Entry type	Description
AD	Auditing changes
AF	Authority failure
AP	Obtaining adopted authority
AU	Attribute changes
AX	Row and column access control
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 Server
DO	Delete object
DS	DST security password reset
EV	System environment variables
GR	Generic record
GS	Socket description was given to another job
IM	Intrusion monitor
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
M0	Db2 Mirror setup tools
M6	Db2 Mirror communication services
M7	Db2 Mirror replication services

Table 161. Audit Journal (QAUDJRN) entry types (continued)	
Entry type	**Description**
M8	Db2 Mirror product services
M9	Db2 Mirror replication state
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
PF	PTF operations
PG	Change of an object's primary group
PO	Printed output
PS	Profile swap
PU	PTF object changes
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	Sockets connections
SM	Systems management changes
SO	Server security user information actions
ST	Use of service tools
SV	System value changed

Table 161. Audit Journal (QAUDJRN) entry types (continued)	
Entry type	**Description**
VA	Changing an access control list (This entry is no longer being written)
VC	Starting or ending a connection (This entry is no longer being written)
VF	Closing server files (This entry is no longer being written)
VL	Account limit exceeded (This entry is no longer being written)
VN	Logging on and off the network (This entry is no longer being written)
VO	Validation list actions
VP	Network password error
VR	Network resource access (This entry is no longer being written)
VS	Starting or ending a server session (This entry is no longer being written)
VU	Changing a network profile (This entry is no longer being written)
VV	Changing service status (This entry is no longer being written)
X0	Network Authentication
X1	Identity Token
X2	Query manager profile changes
XD	Directory server extension
YC	DLO object accessed (change)
YR	DLO object accessed (read)
ZC	Object accessed (change)
ZR	Object accessed (read)

AD (Auditing Change) journal entries

This table provides the format of the AD (Auditing Change) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_AD table function: AUDIT_JOURNAL_AD

Table 162. AD (Auditing Change) journal entries. QASYADJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See “Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)” on page 630,“Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)” on page 632, and “Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)” on page 633 for field listing.

Table 162. AD (Auditing Change) journal entries. QASYADJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
156	224	610	Entry Type	Char(1)	**D** CHGDLOAUD command **O** CHGOBJAUD or CHGAUD command **S** The scan attribute was changed using CHGATR command or the Qp0lSetAttr API, or when the object was created. **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)	If the entry type is D, O, or U, the field contains the current object audit value. If the entry type is S, the field contains the scan attribute value.
					Current audit values:
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.

Table 162. AD (Auditing Change) journal entries. QASYADJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
206	274	660	CHGUSRAUD *SYSMGT	Char(1)	Y = Write an audit record when this user makes systems management changes.
207	275	661	CHGUSRAUD *OPTICAL	Char (1)	Y = Write an audit record when this user accesses optical devices.
		662	CHGUSRAUD *AUTFAIL	Char(1)	Y = Write an audit record when this user has an authorization failure.
		663	CHGUSRAUD *JOBBAS	Char(1)	Y = Write an audit record when this user performs a job base function.
		664	CHGUSRAUD *JOBCHGUSR	Char(1)	Y = Write an audit record when this user changes a thread's active user profile or its group file.
		665	CHGUSRAUD *NETBAS	Char(1)	Y = Write an audit record when this user performs network base functions.
		666	CHGUSRAUD *NETCLU	Char(1)	Y = Write an audit record when this user performs cluster or cluster resource group functions.
		667	CHGUSRAUD *NETCMN	Char(1)	Y = Write an audit record when this user performs network communications functions.
		668	CHGUSRAUD *NETFAIL	Char(1)	Y = Write an audit record when this user has a network failure.
		669	CHGUSRAUD *NETSCK	Char(1)	Y = Write an audit record when this user performs sockets tasks.
		670	CHGUSRAUD *PGMFAIL	Char(1)	Y = Write an audit record when this user has a program failure.
		671	CHGUSRAUD *PRTDTA	Char(1)	Y = Write an audit record when this user performs a print function with parameter SPOOL(*NO).
		672	CHGUSRAUD *SECCFG	Char(1)	Y = Write an audit record when this user performs security configuration.
		673	CHGUSRAUD *SECDIRSRV	Char(1)	Y = Write an audit record when this user makes changes or updates using directory service functions.
		674	CHGUSRAUD *SECIPC	Char(1)	Y = Write an audit record when this user makes changes to interprocess communications.
		675	CHGUSRAUD *SECNAS	Char(1)	Y = Write an audit record when this user performs network authentication service actions.
		676	CHGUSRAUD *SECRUN	Char(1)	Y = Write an audit record when this user performs security run time functions.
		677	CHGUSRAUD *SECSCKD	Char(1)	Y = Write an audit record when this user performs socket descriptor functions.
		678	CHGUSRAUD *SECVFY	Char(1)	Y = Write an audit record when this user uses verification functions.
		679	CHGUSRAUD *SECVLDL	Char(1)	Y = Write an audit record when this user manipulates validation lists.

Table 162. AD (Auditing Change) journal entries. QASYADJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		680	CHGUSRAUD *NETSECURE	Char(1)	Y = Write an audit record when this user establishes a secure connection.
208	276		(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[1]	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[1]	Binary(5)	The coded character set identifier for the path name.
	989	1375	Path Name Country or Region ID[1]	Char(2)	The Country or Region ID for the path name.
	991	1377	Path Name Language ID[1]	Char(3)	The language ID for the path name.

Table 162. AD (Auditing Change) journal entries. QASYADJE/J4/J5 Field Description File *(continued)*					
Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	994	1380	Path Name Length[1]	Binary(4)	The length of the path name.
	996	1382	Path Name Indicator[1]	Char(1)	Path name indicator: **Y** The Path Name field contains complete absolute path name for the object. **N** The Path Name field does not contain an absolute path name for the object, instead it contains a relative path name. The Relative Directory File ID field is valid and can be used to form an absolute path name with this relative path name.
	997	1383	Relative Directory File ID[1, 3]	Char(16)	When the Path Name Indicator field is N, this field contains the file ID of the directory that contains the object identified in the Path Name field. Otherwise it contains hex zeros.[3]
	1013	1399	Path Name[1, 4]	Char(5002)	The path name of the object.
		6401	Previous Object Audit Value	Char(10)	If the entry type is D, O, or U, the field contains the previous audit value.
					Previous audit values:
		6411	CHGUSRAUD *CMD	Char(1)	Y = Audit commands for this user.
		6412	CHGUSRAUD *CREATE	Char(1)	Y = Write an audit record when this user creates an object.
		6413	CHGUSRAUD *DELETE	Char(1)	Y = Write an audit record when this user deletes an object.
		6414	CHGUSRAUD *JOBDTA	Char(1)	Y = Write an audit record when this user changes a job.
		6415	CHGUSRAUD *OBJMGT	Char(1)	Y = Write an audit record when this user moves or renames an object.
		6416	CHGUSRAUD *OFCSRV	Char(1)	Y = Write an audit record when this user performs office functions.
		6417	CHGUSRAUD *PGMADP	Char(1)	Y = Write an audit record when this user obtains authority through adopted authority.
		6418	CHGUSRAUD *SAVRST	Char(1)	Y = Write an audit record when this user saves or restores objects.
		6419	CHGUSRAUD *SECURITY	Char(1)	Y = Write an audit record when this user performs security-relevant actions.
		6420	CHGUSRAUD *SERVICE	Char(1)	Y = Write an audit record when this user performs service functions.

Table 162. AD (Auditing Change) journal entries. QASYADJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		6421	CHGUSRAUD *SPLFDTA	Char(1)	Y = Write an audit record when this user manipulates spooled files.
		6422	CHGUSRAUD *SYSMGT	Char(1)	Y = Write an audit record when this user makes system management changes.
		6423	CHGUSRAUD *OPTICAL	Char(1)	Y = Write an audit record when this user accesses optical devices.
		6424	CHGUSRAUD *AUTFAIL	Char(1)	Y = Write an audit record when this user has an authorization failure.
		6425	CHGUSRAUD *JOBBAS	Char(1)	Y = Write an audit record when this user performs a job base function.
		6426	CHGUSRAUD *JOBCHGUSR	Char(1)	Y = Write an audit record when this user changes a thread's active user profile.
		6427	CHGUSRAUD *NETBAS	Char(1)	Y = Write an audit record when this user performs network base functions.
		6428	CHGUSRAUD *NETCLU	Char(1)	Y = Write an audit record when this user performs cluster or cluster resource group functions.
		6429	CHGUSRAUD *NETCMN	Char(1)	Y = Write an audit record when this user performs network communications functions.
		6430	CHGUSRAUD *NETFAIL	Char(1)	Y = Write an audit record when this user has a network failure.
		6431	CHGUSRAUD *NETSCK	Char(1)	Y = Write an audit record when this user performs sockets tasks.
		6432	CHGUSRAUD *PGMFAIL	Char(1)	Y = Write an audit record when this user has a program failure.
		6433	CHGUSRAUD *PRTDTA	Char(1)	Y = Write an audit record when this user performs a print function with parameter SPOOL(*NO)
		6434	CHGUSRAUD *SECCFG	Char(1)	Y = Write an audit record when this user performs security configuration.
		6435	CHGUSRAUD *SECDIRSRV	Char(1)	Y = Write an audit record when this user makes changes or updates using directory service functions.
		6436	CHGUSRAUD *SECIPC	Char(1)	Y = Write an audit record when this user makes changes to interprocess communications.
		6437	CHGUSRAUD *SECNAS	Char(1)	Y = Write an audit record when this user performs network authentication service actions.
		6438	CHGUSRAUD *SECRUN	Char(1)	Y = Write an audit record when this user performs security run time functions.
		6439	CHGUSRAUD *SECSCKD	Char(1)	Y = Write an audit record when this user performs socket descriptor functions.
		6440	CHGUSRAUD *SECVFY	Char(1)	Y = Write an audit record when this user uses verification functions.

Table 162. AD (Auditing Change) journal entries. QASYADJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		6441	CHGUSRAUD *SECVLDL	Char(1)	Y = Write an audit record when this user manipulates validation lists.
		6442	CHGUSRAUD *NETSECURE	Char(1)	Y = Write an audit record when this user establishes a secure connection.
		6443	CHGUSRAUD *NETUDP	Char(1)	Y = Write an audit record for UDP inbound and outbound traffic for this user.
					End of previous audit values
		6444	Reserved	Char(10)	Not used
		6454	CHGUSRAUD *NETUDP	Char(1)	Current audit value. Y = Write an audit record for UDP inbound and outbound traffic for this user.

1
These fields are used only for objects in the "root" (/), QOpenSys, 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
If the Path Name Indicator field is N, but the Relative Directory File ID is hex zeros, then there was some error in determining the path name information.

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.

AF (Authority Failure) journal entries

This table provides the format of the AF (Authority Failure) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_AF table function: AUDIT_JOURNAL_AF

Table 163. AF (Authority Failure) journal entries. QASYAFJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See “Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)” on page 630,“Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)” on page 632, and “Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)” on page 633 for field listing.

Table 163. AF (Authority Failure) journal entries. QASYAFJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
156	224	610	Violation Type[1]	Char(1)	**A** Not authorized to object **B** Restricted instruction **C** Validation failure (see J5 offset 639) **D** Use of unsupported interface, object domain failure **E** Hardware storage protection error, program constant space violation **F** ICAPI authorization error (obsolete) **G** ICAPI authentication error (obsolete) **H** Scan exit program action (see J5 offset 639) **I**[7] System Java inheritance not allowed **J** Submit job profile error **K** Special authority violation **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

Table 163. AF (Authority Failure) journal entries. QASYAFJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
					(continued) **V** Profile token not valid for generating new profile token **W** Profile token not valid for swap **X** System violation — see J5 offset 723 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, 12, 17]	Char(10)	The name of the object.
167	235	621	Library Name[13]	Char(10)	The name of the library where the object is stored or the Licensed Internal Code fix number that failed to apply.[11]
177	245	631	Object Type[14,17]	Char(8)	The type of object.

Table 163. AF (Authority Failure) journal entries. QASYAFJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
185	253	639	Validation Error Action	Char(1)	Action taken after validation error detected, set only if the violation type (J5 offset 610) is C or H. **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 IBM i format. **G** Unsigned system or inherit state object found when checking system. **H** Unsigned user state object found when checking system. **I** Mismatch between object and its signature found when checking system. **J** IBM certificate not found when checking system.

Table 163. AF (Authority Failure) journal entries. QASYAFJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
					(continued) **K** Invalid signature format found when checking system. **M** Scan exit program modified the object that was scanned **X** Scan exit program wanted object marked as having a scan failure
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	Workstation Name	Char(10)	The name of the workstation or workstation 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 163. AF (Authority Failure) journal entries. QASYAFJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
269	337	723	Operation Violation Code	Char(3)	The type of operation violation that occurred, set only if the violation type (J5 offset 610) is X. **AAC** Not authorized to use SST Advanced Analysis Command. **HCA** Service tool user profile not authorized to perform hardware configuration operation (QYHCHCOP). **LIC** LIC indicates that a Licensed Internal Code fix was not applied because of a signature violation. **SFA** Not authorized to activate the environment attribute for system file access. **CMD** An attempt was made to use a command that has been disabled by a system administrator.
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[15, 16]	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)	

Table 163. AF (Authority Failure) journal entries. QASYAFJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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[3]	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[3]	Binary(5)	The coded character set identifier for the path name.
	1054	1440	Path Name Country or Region ID[3]	Char(2)	The Country or Region ID for the path name.
	1056	1442	Path Name Language ID[3]	Char(3)	The language ID for the path name.
	1059	1445	Path Name Length[3]	Binary(4)	The length of the path name.
	1061	1447	Path Name Indicator[3]	Char(1)	Path name indicator: **Y** The Path Name field contains complete absolute path name for the object. **N** The Path Name field does not contain an absolute path name for the object, instead it contains a relative path name. The Relative Directory File ID field is valid and can be used to form an absolute path name with this relative path name.
	1062	1448	Relative Directory File ID[3, 8]	Char(16)	When the Path Name Indicator field is N, this field contains the file ID of the directory that contains the object identified in the Path Name field. Otherwise it contains hex zeros.[8]
	1078	1464	Path Name[3, 9]	Char(5002)	The 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

Table 163. AF (Authority Failure) journal entries. QASYAFJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
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 "Security auditing journal entries" on page 272.
2					This field contains the name of the user that caused the entry. QSYS might be the user for the following entries: • 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 "root" (/), QOpenSys, and user-defined file systems.
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 integrated file system 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 can not extend its base class because the base class has system Java attributes.
8					If the Path Name Indicator field is N, but the Relative Directory File ID is hex zeros, then there was some error in determining the path name information.
9					This is a variable length field. The first two 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 163. AF (Authority Failure) journal entries. QASYAFJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	J4	J5			
11 When the violation type is X and the Operation Violation code value is LIC, this indicates that a Licensed Internal Code fix was not applied because of a signature violation. This field will contain the Licensed Internal Code fix number that failed to apply. 12 When the violation type is K, the object name contains the name of the command or program that detected the error. If the command has several alternative names, the command name in the audit record might not match the specific command name used but will be one of the equivalent alternatives. A special value of *INSTR indicates that a machine instruction detected the error. 13 When the violation type is K, the library name contains the name of the program's library or *N for the command's library that detected the error. 14 When the violation type is K, the object type contains the object type of the command or program that detected the error. 15 When the violation type is K, the Folder Path might contain the full API name of the API or exit point name that detected the error.					
16 When the violation type is X and the Operation Violation Code is AAC, the Folder Path will contain the 30 character Advanced Analysis Command name. 17 When the object type is *LIC and the object library is *N, the object name is a Licensed Internal Code Ru name.					

AP (Adopted Authority) journal entries

This table provides the format of the AP (Adopted Authority) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_AP table function: AUDIT_JOURNAL_AP

Table 164. AP (Adopted Authority) journal entries. QASYAPJE/J4/J5 Field Description File

Offset			Field	Format	Description
JE	J4	J5			
1	1	1			Heading fields common to all entry types. See “Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)” on page 630,“Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)” on page 632, and “Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)” on page 633 for field listing.

Table 164. AP (Adopted Authority) journal entries. QASYAPJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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.
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.					

AU (Attribute Changes) journal entries

This table provides the format of the AU (Attribute Changes) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_AU table function: AUDIT_JOURNAL_AU

Table 165. AU (Attribute Changes) journal entries. QASYAUJ5 Field Description File			
Offset			
J5	**Field**	**Format**	**Description**
610	Entry type	Char(1)	The type of entry. **E** EIM configuration attributes **A** EIM association

Table 165. AU (Attribute Changes) journal entries. QASYAUJ5 Field Description File *(continued)*			
Offset			
J5	**Field**	**Format**	**Description**
611	Action	Char(3)	Action. When entry type (J5 offset 610) is E this field can contain: **CHG** Attributes changed When entry type (J5 offset 610) is A this field can contain: **ADD** Add association **RMV**[2] Remove association
614	Name	Char(100)	Attribute name. When entry type (J5 offset 610) is A this field contains the registry user name.
714	New Value Length	Binary(4)	New value length. When entry type (J5 offset 610) is A this field contains the length of the identifier dn.
716	New Value CCSID	Binary(5)	New value CCSID. When entry type (J5 offset 610) is A this field contains the CCSID of the identifier dn.
720	New Value Country or Region ID	Char(2)	New value Country or Region ID. When entry type (J5 offset 610) is A this field contains the Country or Region ID of the identifier dn.
722	New Value Language ID	Char(3)	New value language ID. When entry type (J5 offset 610) is A this field contains the language ID of the identifier dn.
725	New Value	Char(2002) [1]	New value. When entry type (J5 offset 610) is A this field contains the identifier dn.
2727	Old Value Length	Binary(4)	Old value length. When entry type (J5 offset 610) is A this field contains the length of the registry dn.
2729	Old Value CCSID	Binary(5)	Old value CCSID. When entry type (J5 offset 610) is A this field contains the CCSID of the registry dn.
2733	Old Value Country or Region ID	Char(2)	Old value Country or Region ID. When entry type (J5 offset 610) is A this field contains the Country or Region ID of the registry dn.

Table 165. AU (Attribute Changes) journal entries. QASYAUJ5 Field Description File *(continued)*			
Offset J5	**Field**	**Format**	**Description**
2735	Old Value Language ID	Char(3)	Old value language ID. When entry type (J5 offset 610) is A this field contains the language ID of the registry dn.
2738	Old Value	Char(2002) [1]	Old value. When entry type (J5 offset 610) is A this field contains the registry dn.
4740	Association Type	Char(1)	When entry type (J5 offset 610) is A this field contains the association type being added or removed. **0** All **1** Target **2** Source **3** Source and target **4** Administrative

1 This is a variable length field. The first two bytes contain the length of the field.

2 A remove association audit entry is not sent when the remove association is a result of the removal of a registry or the removal of an identifier.

AX (Row and Column Access Control) journal entries

This table provides the format of the AX (Row and Column Access Control) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_AX table function: AUDIT_JOURNAL_AX

Table 166. AX (Row and Column Access Control) journal entries. QASYAXJ5 Field Description File			
Offset J5	**Field**	**Format**	**Description**
1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630 for field listing.

Table 166. AX (Row and Column Access Control) journal entries. QASYAXJ5 Field Description File *(continued)*			
Offset			
J5	**Field**	**Format**	**Description**
610	Entry Type	Char(1)	The type of entry. **M** Column mask **P** Row permission **T** Table
611	Operation Type	Char(1)	The type of operation. **A** Alter **B** Internal use **C** Create **D** Drop
612	Table Name	Char(10)	The name of the base table that the permission or mask is associated with or the table being altered.
622	Table Library	Char(10)	The name or the library where the table is stored.
632	Table ASP Name	Char(10)	The name of the table ASP device.
642	Table ASP Number	Char(5)	The number of the table ASP device.
647	Name	Char(128)	When entry type (J5 offset 610) is P this field contains the row permission name. When entry type (J5 offset 610) is M this field contains the column mask name.
775	Column Name	Char(10)	The name of the column to which the mask applies. This field is only used when the entry type (J5 offset 610) is M and the operation type (J5 offset 611) is C.

Table 166. AX (Row and Column Access Control) journal entries. QASYAXJ5 Field Description File *(continued)*			
Offset			
J5	**Field**	**Format**	**Description**
785	Status 1	Char(1)	This field is only used when the operation type (J5 offset 611) is A or C. When entry type (J5 offset 610) is M or P this field contains the row permission status or the column mask status. **E** Enabled **D** Disabled When entry type (J5 offset 610) is T this field contains the row access control status. **A** Activate **D** Deactivate
786	Status 2	Char(1)	This field is only used when the operation type (J5 offset 611) is A. When entry type (J5 offset 610) is T this field contains the column access control status. **A** Activate **D** Deactivate
787	Previous Status 1	Char(1)	This field is only used when the operation type (J5 offset 611) is A. When entry type (J5 offset 610) is M or P this field contains the previous row permission status or the previous column mask status. **E** Enabled **D** Disabled When entry type (J5 offset 610) is T this field contains the previous row access control status. **A** Activate **D** Deactivate

Table 166. AX (Row and Column Access Control) journal entries. QASYAXJ5 Field Description File *(continued)*			
Offset			
J5	**Field**	**Format**	**Description**
788	Previous Status 2	Char(1)	This field is only used when the operation type (J5 offset 611) is A. When entry type (J5 offset 610) is T this field contains the previous column access control status. **A** Activate **D** Deactivate
789	(Reserved Area)	Char(50)	
839	Truncated Indicator	Char(1)	Indicates if the SQL statement is truncated. This field is only used when the entry type (J5 offset 610) is M or P and the operation type (J5 offset 611) is C. **1** SQL statement truncated
840	SQL statement CCSID	Binary(5)	The coded character set identifier for the SQL statement. This field is only used when the entry type (J5 offset 610) is M or P and the operation type (J5 offset 611) is C.
844	SQL statement length	Binary(4)	The length of the SQL statement. This field is only used when the entry type (J5 offset 610) is M or P and the operation type (J5 offset 611) is C.
846	SQL statement[1]	Char(5002)	The SQL statement. This field is only used when the entry type (J5 offset 610) is M or P and the operation type (J5 offset 611) is C.
1 This is a variable length field. The first two bytes contain the length of the SQL statement.			

CA (Authority Changes) journal entries

This table provides the format of the CA (Authority Changes) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_CA table function: AUDIT_JOURNAL_CA

Table 167. 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 "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 for field listing.

Table 167. CA (Authority Changes) journal entries. QASYCAJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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 where the object is stored.
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:
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

Table 167. CA (Authority Changes) journal entries. QASYCAJE/J4/J5 Field Description File *(continued)*					
Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
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		(Reserved Area)	Char(10)	
		689	Object Attribute	Char(10)	The attribute of the object.
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.
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.

Table 167. CA (Authority Changes) journal entries. QASYCAJE/J4/J5 Field Description File *(continued)*					
Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
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 path name.
	1033	1419	Path Name Country or Region ID	Char(2)	The Country or Region ID for the path name.
	1035	1421	Path Name Language ID	Char(3)	The language ID for the path name.
	1038	1424	Path Name Length	Binary(4)	The length of the path name.
	1040	1426	Path Name Indicator	Char(1)	Path name indicator: **Y** The Path Name field contains complete absolute path name for the object. **N** The Path Name field does not contain an absolute path name for the object, instead it contains a relative path name. The Relative Directory File ID field is valid and can be used to form an absolute path name with this relative path name.
	1041	1427	Relative Directory File ID[3]	Char(16)	When the Path Name Indicator field is N, this field contains the file ID of the directory that contains the object identified in the Path Name field. Otherwise it contains hex zeros.[3]
	1057	1443	Path Name[4]	Char(5002)	The path name of the object.

Table 167. CA (Authority Changes) journal entries. QASYCAJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		6445	Previous Authorization List Name	Char(10)	The name of the previous authorization list.
					Previous authorities
		6455	Previous Object Existence	Char(1)	**Y** *OBJEXIST
		6456	Previous Object Management	Char(1)	**Y** *OBJMGT
		6457	Previous Object Operational	Char(1)	**Y** *OBJOPR
		6458	Previous Authorization List Management	Char(1)	**Y** *AUTLMGT
		6459	Previous Authorization List Authority	Char(1)	**Y** *AUTL public authority
		6460	Previous Read Authority	Char(1)	**Y** *READ
		6461	Previous Add Authority	Char(1)	**Y** *ADD
		6462	Previous Update Authority	Char(1)	**Y** *UPD
		6463	Previous Delete Authority	Char(1)	**Y** *DLT
		6464	Previous Exclude Authority	Char(1)	**Y** *EXCLUDE[6]
		6465	Previous Execute Authority	Char(1)	**Y** *EXECUTE
		6466	Previous Object Alter Authority	Char(1)	**Y** *OBJALTER
		6467	Previous Object Reference Authority	Char(1)	**Y** *OBJREF

Table 167. CA (Authority Changes) journal entries. QASYCAJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1 These fields are used only for objects in the "root" (/), QOpenSys, 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** If the Path Name Indicator field is N, but the Relative Directory File ID is hex zeros, then there was some error in determining the path name information. **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. **6** New objects may show a previous authority of *EXCLUDE because of the way in which the system assigns authorities to new objects.					

CD (Command String) journal entries

This table provides the format of the CD (Command String) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_CD table function: AUDIT_JOURNAL_CD

Table 168. CD (Command String) journal entries. QASYCDJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 for field listing.

Table 168. CD (Command String) journal entries. QASYCDJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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 **X** Proxy command
157	225	611	Object Name	Char(10)	The name of the object.
167	235	621	Library Name	Char(10)	The name of the library where the object is stored.
177	245	631	Object Type	Char(8)	The type of object.
185	253	639	Where run	Char(1)	Where the CL command was run. **Y** From a compiled OPM CL program or an ILE CL Program **R** From a REXX procedure **E** The command string was passed as a parameter to one of the Command Analyzer APIs: QCMDEXC, QCAPCMD, or QCAEXEC **B** In a batch job but not for any of the reason listed under Y, R, or E. Typical case would be that the CL command was run using STRDBRDR or SBMDBJOB command or was specified on the CMD parameter of the SBMJOB command. **N** Interactively from a command line or by choosing a menu option that runs a CL command
186	254	640	Command String	Char(6000)	The command that was run, with parameters.

Table 168. CD (Command String) journal entries. QASYCDJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		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

CO (Create Object) journal entries

This table provides the format of the CO (Create Object) journal entries. Objects created into QTEMP library are not audited.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_CO table function: AUDIT_JOURNAL_CO

Table 169. CO (Create Object) journal entries. QASYCOJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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		(Reserved Area)	Char(20)	
		639	Object Attribute	Char(10)	The attribute of the object.
		649	(Reserved Area)	Char(10)	
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.

Table 169. CO (Create Object) journal entries. QASYCOJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
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	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 path name.
	987	1373	Path Name Country or Region ID	Char(2)	The Country or Region ID for the path name.
	989	1375	Path Name Language ID	Char(3)	The language ID for the path name.
	992	1378	Path Name Length	Binary(4)	The length of the path name.

Table 169. CO (Create Object) journal entries. QASYCOJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	994	1380	Path Name Indicator	Char(1)	Path name indicator: **Y** The Path Name field contains complete absolute path name for the object. **N** The Path Name field does not contain an absolute path name for the object, instead it contains a relative path name. The Relative Directory File ID field is valid and can be used to form an absolute path name with this relative path name.
	995	1381	Relative Directory File ID[3]	Char(16)	When the Path Name Indicator field is N, this field contains the file ID of the directory that contains the object identified in the Path Name field. Otherwise it contains hex zeros.[3]
	1011	1397	Path Name[4]	Char(5002)	The path name of the object.

1
These fields are used only for objects in the "root" (/), QOpenSys, 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
If the Path Name Indicator field is N, but the Relative Directory File ID is hex zeros, then there was some error in determining the path name information.

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.

CP (User Profile Changes) journal entries

This table provides the format of the CP (User Profile Changes) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_CP table function: AUDIT_JOURNAL_CP

Table 170. CP (User Profile Changes) journal entries. QASYCPJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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 or CHGEXPSCDE **RST** RSTUSRPRF **DST** QSECOFR password reset using DST **RPA** QSYRESPA API **SQL** QSYS2/SET_SERVER_SBS_ROUTING() Db2 for i procedure
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** Current *ALLOBJ special authority

Table 170. CP (User Profile Changes) journal entries. QASYCPJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
192	260	646	Job Control Special Authority	Char(1)	**Y** Current *JOBCTL special authority
193	261	647	Save System Special Authority	Char(1)	**Y** Current *SAVSYS special authority
194	262	648	Security Administrator Special Authority	Char(1)	**Y** Current *SECADM special authority
195	263	649	Spool Control Special Authority	Char(1)	**Y** Current *SPLCTL special authority
196	264	650	Service Special Authority	Char(1)	**Y** Current *SERVICE special authority
197	265	651	Audit Special Authority	Char(1)	**Y** Current *AUDIT special authority
198	266	652	System Configuration Special Authority	Char(1)	**Y** Current *IOSYSCFG special authority
199	267		(Reserved Area)	Char(13)	
		653	Previous All Object Special Authority	Char(1)	**Y** Previous *ALLOBJ special authority
		654	Previous Job Control Special Authority	Char(1)	**Y** Previous *JOBCTL special authority
		655	Previous Save System Special Authority	Char(1)	**Y** Previous *SAVSYS special authority
		656	Previous Security Administrator Special Authority	Char(1)	**Y** Previous *SECADM special authority
		657	Previous Spool Control Special Authority	Char(1)	**Y** Previous *SPLCTL special authority

Table 170. CP (User Profile Changes) journal entries. QASYCPJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		658	Previous Service Special Authority	Char(1)	**Y** Previous *SERVICE special authority
		659	Previous Audit Special Authority	Char(1)	**Y** Previous *AUDIT special authority
		660	Previous System Configuration Special Authority	Char(1)	**Y** Previous *IOSYSCFG special authority
		661	(Reserved Area)	Char(5)	
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.

Table 170. CP (User Profile Changes) journal entries. QASYCPJE/J4/J5 Field Description File *(continued)*					
Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
493	561	947	Group Identification	Char(10)	The gid for the user.
503	571	957	Local Password Management	Char(10)	The value of the LCLPWDMGT parameter.
		967	Password Composition Conformance	Char(10)	Indicates whether the new password conforms to the password composition rules. ***PASSED** Checked and conforms. ***SYSVAL** Checked but does not conform because of a system value based rule. ***EXITPGM** Checked but does not conform because of an exit program response. ***NONE** Not checked; *NONE was specified for the new password. ***NOCHECK** Not checked; password was changed. This field has meaning only when the Password Changed field contains a *Y*.
		977	Password Expiration Interval	Char(7)	Specifies the value that the password expiration interval has been changed to. ***NOMAX** No expiration interval. ***SYSVAL** The system value QPWDEXPITV is used. **number** The size of the expiration interval in days.
		984	Block Password Change	Char(10)	Specifies the value that the block password change has been changed to. ***SYSVAL** The system value QPWDCHGBLK is used. ***NONE** No block period. **1-99** Blocked hours.
		994	User Expiration Date	Char(7)	Specifies the date when the user profile expires (CYYMMDD). The user profile is automatically disabled or deleted on this date.

Table 170. CP (User Profile Changes) journal entries. QASYCPJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		1001	Alternative Subsystem Name	Char(10)	The alternative subsystem that will be used for this user, instead of the default subsystem, whenever a connection is initiated to the server job specified in the IBM i Server Job Name field. This field will only contain data when command name (J5 offset 639) is SQL.
		1011	IBM i Server Job Name	Char(10)	When a connection to this server is initiated for this user it will be routed to the subsystem specified in the Alternative Subsystem Name field. To understand the Server Job Name mapping to server names and the default subsystem use, see Server table. This field will only contain data when command name (J5 offset 639) is SQL.
		1021	Assistance Level	Char(10)	The user interface that will be used. ***SYSVAL** The system value, QASTLVL, is used to determine the user interface that will be used. ***BASIC** The Operational Assistant user interface is used. ***INTERMED** The system interface is used. ***ADVANCED** The expert system interface is used.
		1031	Special Environment	Char(10)	The special environment in which the user operates after signing on. ***SYSVAL** The system value, QSPCENV, is used to determine the system environment in which the user operates after signing on the system. ***NONE** The user operates in the IBM i system environment after signing on the system. ***S36** The user operates in the System/36 environment after signing on the system.

Table 170. CP (User Profile Changes) journal entries. QASYCPJE/J4/J5 Field Description File *(continued)*					
Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
		1041	Display Signon Information	Char(10)	Indicates if the sign-on information display is shown. ***SYSVAL** The system value, QDSPSGNINF, is used to determine whether the sign-on information display is shown. ***NO** The sign-on information display is not shown. ***YES** The sign-on information display is shown.
		1051	Limit Device Sessions	Char(10)	The number of device sessions allowed for a user is limited. ***SYSVAL** The system value, QLMTDEVSSN, is used to determine whether the user is limited to a specific number of device sessions. ***NO** The user is not limited to a specific number of device sessions. ***YES** The user is limited to a single device session. **0** The user is not limited to a specific number of device sessions. This value has the same meaning as *NO. **1** The user is no limited to a single device sessions. This value has the same meaning as *YES. **2-9** The user is limited to the specified number of device sessions.
		1061	Keyboard Buffering	Char(10)	The keyboard buffering value to be used when a job is initialized for this user profile. ***SYSVAL** The system value, QKBDBUF, is used to determine the keyboard buffering value. ***NO** The type-ahead feature and attention key buffering option are not active. ***TYPEAHEAD** The type-ahead feature is active, but the attention key buffering option is not. ***YES** The type-ahead feature and attention key buffering option are active.

Table 170. CP (User Profile Changes) journal entries. QASYCPJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		1071	Maximum Allowed Storage	Char(20)	The amount of auxiliary storage (in kilobytes) assigned to store permanent objects owned by this user profile in the system auxiliary storage pool (ASP) and on all the basic ASPs combined. In addition, this value also controls the maximum amount of auxiliary storage that can be used to store permanent objects owned by this user profile on each Independent ASP (IASP).
		1091	Job Description	Char(10)	The job description used for jobs that start through subsystem work station entries whose job description parameter values indicate the user JOBD(*USRPRF).
		1101	Job Description Library	Char(10)	The name of the library where the job description is found.
		1111	Accounting Code	Char(15)	The accounting code that is associated with this user profile or the value listed below. ***BLANK** An accounting code of 15 blanks is assigned to this user profile.
		1126	Document Password Changed	Char(1)	Indicates if the document password has been changed. **Y** Document password changed.
		1127	Document Password *NONE	Char(1)	Indicates if the document password is *NONE. **Y** Document password is *NONE.
		1128	Message Queue	Char(10)	The message queue to which messages are sent or the value listed below. ***USRPRF** A message queue with the same name as the user profile is used as the message queue for this user. The message queue is located in the QUSRSYS library.
		1138	Message Queue Library	Char(10)	The name of the library where the message queue is found.

Table 170. CP (User Profile Changes) journal entries. QASYCPJE/J4/J5 Field Description File *(continued)*					
Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
		1148	Delivery	Char(10)	How messages sent to the message queue for this user are to be delivered. ***NOTIFY** The job to which the message queue is assigned is notified when a message arrives at the message queue. ***HOLD** The messages are held in the message queue until they are requested by the user or program. ***BREAK** The job to which the message queue is assigned is interrupted when a message arrives at the message queue. ***DFT** The default reply to the inquiry message is sent.
		1158	Severity Code Filter	Char(2)	The lowest severity code that a message can have and still be delivered to a user in break or notify mode. **00-99**
		1160	Print Device	Char(10)	The default printer device for this user or one of the values listed below. ***WRKSTN** The printer assigned to the user's work station is used. ***SYSVAL** The system value, QPRTDEV, is used to determine the printer device.
		1170	Output Queue	Char(10)	The output queue to be used by this user profile or one of the values listed below. ***WRKSTN** The output queue assigned to the user's work station is used. ***DEV** The output queue associated with the printer specified for the Printer Device is used.
		1180	Output Queue Library	Char(10)	The name of the library where the output queue is found.

Table 170. CP (User Profile Changes) journal entries. QASYCPJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		1190	Attention Program	Char(10)	The program to be used as the Attention (ATTN) key handling program for this user or one of the values listed below. ***SYSVAL** The system value, QATNPGM, is used to determine the ATTN key handling program. ***NONE** No ATTN key handling program is used by this user. ***ASSIST** The Operational Assistant ATTN key handling program, QEZMAIN, is used.
		1200	Attention Program Library	Char(10)	The name of the library where the ATTN program is found.
		1210	Sort Sequence	Char(10)	The sort sequence table to be used for string comparisons for this user profile or one of the values listed below. ***SYSVAL** The system value, QSRTSEQ, is used to determine the sort sequence table. ***HEX** A sort sequence table is not used. The hexadecimal values of the characters are used to determine the sort sequence. ***LANGIDUNQ** A unique-weight sort table is used. ***LANGIDSHR** A shared-weight sort table is used.
		1220	Sort Sequence Library	Char(10)	The name of the library where the sort sequence table is found.
		1230	Language ID	Char(10)	The language identifier to be used for this user profile or the value listed below. ***SYSVAL** The system value, QLANGID, is used to determine the language identifier.
		1240	Country or Region ID	Char(10)	The country or region identifier to be used for this user profile or the value listed below. ***SYSVAL** The system value, QCNTRYID, is used to determine the country or region ID.
		1250	CCSID	Binary(5)	The coded character set identifier to be used for this user profile.

Table 170. CP (User Profile Changes) journal entries. QASYCPJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		1254	Character Identifier Control	Char(10)	The character identifier control (CHRIDCTL) for the job. ***SYSVAL** The system value, QCHRIDCTL, is used to determine the CHRIDCTL for the job. ***DEVD** Performs the same function as it does on the CHRID parameter for display files, printer files, and panel groups. ***JOBCCSID** Performs the same function as it does on the CHRID parameter for display files, printer files, and panel groups.
		1264	Locale Job Attributes	Char(60)	The job attributes that are to be taken from the locale when the job is initiated. This field can contain up to six char(10) values. ***SYSVAL** The system value, QSETJOBATR, is used to determine which job attributes are taken from the locale. ***NONE** No job attributes are taken from the locale. ***CCSID** The coded character set identifier from the locale is used. ***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. ***TIMSEP** The time separator from the locale is used.

Table 170. CP (User Profile Changes) journal entries. QASYCPJE/J4/J5 Field Description File *(continued)*					
Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
		1324	User Options	Char(70)	The level of help information detail to be shown and the default function of the Page Up and Page Down keys. This field can contain up to seven char(10) values. ***NONE** Detailed information is not shown. ***CLKWD** Parameter keywords are shown instead of the possible parameter values when a control language (CL) command is prompted. ***EXPERT** More detailed information is shown when the user is performing display and edit options to define or change the system. ***ROLLKEY** The actions of the Page Up and Page Down keys are reversed. ***NOSTSMSG** Status messages are not displayed when sent to the user. ***STSMSG** Status messages are displayed when sent to the user. ***HLPFULL** Help text is shown on a full display rather than in a window. ***PRTMSG** A message is sent to this user's message queue when a spooled file for this user is printed or held by the printer writer.
		1394	EIM Identifier	Char(128)	Enterprise Identity Mapping (EIM) identifier name or the value listed below. ***USRPRF** The name of the EIM identifier is the same name as the user profile.
		1522	EIM Association Type	Char(10)	EIM association type. ***TARGET** Target association. ***SOURCE** Source association. ***TGTSRC** Target and source associations. ***ADMIN** Administrative association. ***ALL** All association types.

Table 170. CP (User Profile Changes) journal entries. QASYCPJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		1532	EIM Association Action	Char(10)	EIM association action. ***REPLACE** Associations of the specified type will be removed from all EIM identifiers that have an association for this user profile and local EIM registry. A new association will be added to the specified EIM identifier. ***ADD** Add an association. ***REMOVE** Remove an association.
		1542	Create EIM Identifier	Char(12)	Indicates whether the EIM identifier should be created if it does not exist. ***NOCRTEIMID** EIM identifier does not get created. ***CRTEIMID** EIM identifier gets created if it does not exist.
		1554	User Expiration Action	Char(3)	The action performed on the profile when it expires. This value is always DSB when using the CRTUSRPRF and CHGUSRPRF commands. When using the CHGEXPSCDE command, this value is one of the value listed below. **DSB** The profile is disabled when it expires. **DLT** The profile is deleted when it expires.
		1557	Owned Object Option Value	Char(1)	The type of operation performed on the objects owned by the expiring profile when the user expiration action (J5 offset 1554) is DLT. The owned object option value is specified on the OWNOBJOPT parameter of the CHGEXPSCDE ACTION(*DELETE) command. **N** *NODLT - The owned objects for the user profile are not changed, and the user profile is not deleted if the user owns any objects. **D** *DLT - The owned objects for the user profile are deleted. The user profile is deleted if the deletion of all owned objects is successful. **C** *CHGOWN - The owned objects for the user profile have ownership transferred to the new owner user profile. The user profile is deleted if the transfer of all owned objects is successful.

Table 170. CP (User Profile Changes) journal entries. QASYCPJE/J4/J5 Field Description File *(continued)*					
Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
		1558	Owned Object Option New Owner	Char(10)	The profile that will own all of the objects owned by the expiring profile. This field will only contain data when the owned object option value (J5 offset 1557) is C.
		1568	Primary Group Option Value	Char(1)	The type of operation performed on the objects that have the expiring user profile as their primary group when the user expiration action (J5 offset 1554) is DLT. The primary group option value is specified on the PGPOPT parameter of the CHGEXPSCDE ACTION(*DELETE) command. **N** *NOCHG - The objects the user profile is the primary group for do not change, and the user profile is not deleted if the user is the primary group for any objects. **C** *CHGPGP - The objects the user profile is the primary group for are transferred to the new primary group user profile. The user profile is deleted if the transfer of all objects is successful.
		1569	Primary Group Option New Primary Group	Char(10)	The profile that will become the new primary group of the objects for which the expiring profile is the primary group. This field will only contain data when the primary group option value (J5 offset 1568) is C. This field may contain the value listed below. ***NONE** All of the objects for which the expiring user is the primary group will no longer have a primary group.

Table 170. CP (User Profile Changes) journal entries. QASYCPJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		1579	Primary Group Option New Primary Group Authority	Char(1)	The authority the new primary group has to the object. This field will only contain data when the primary group option value (J5 offset 1568) is C and the new primary group (J5 offset 1569) is not *NONE. **O** *OLDPGP - The new primary group has the same authority to the object as the old primary group. **P** *PRIVATE - The new primary group has the same authority to the object as its private authority to the object was. **A** *ALL - The new primary group has *ALL authority to the object. **C** *CHANGE - The new primary group has *CHANGE authority to the object. **U** *USE - The new primary group has *USE authority to the object. **E** *EXCLUDE - The new primary group has *EXCLUDE authority to the object.
		1580	(Reserved Area)	Char(26)	
		1606	Home Directory CCSID	Binary(5)	The coded character set identifier for the home directory.
		1610	Home Directory Length	Binary(4)	Length of the home directory.
		1612	Home Directory[1]	Char(5002)	Path name of the home directory or the value listed below. ***USRPRF** The home directory assigned to the user will be /home/USRPRF, where USRPRF is the name of the user profile. For this value, the length will be 7 and the CCSID will be 37.
		6614	Locale CCSID	Binary(5)	The coded character set identifier for the locale.
		6618	Locale Length	Binary(4)	Length of the locale.

Table 170. CP (User Profile Changes) journal entries. QASYCPJE/J4/J5 Field Description File *(continued)*					
Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
		6620	Locale[1]	Char(5002)	Path name of the locale or one of the values listed below. ***SYSVAL** The system value, QLOCALE, is used to determine the locale path name to be assigned to this user. For this value, the length will be 7 and the CCSID will be 37. ***NONE** No locale path name is assigned to this user. For this value, the length will be 5 and the CCSID will be 37. ***C** The C locale path name is assigned to this user. For this value, the length will be 2 and the CCSID will be 37. ***POSIX** The POSIX locale path name is assigned to this user. For this value, the length will be 6 and the CCSID will be 37.
1 This is a variable length field. The first two bytes contain the length of the path name.					

CQ (*CRQD Changes) journal entries

This table provides the format of the CQ (*CRQD Changes) journal entries.

*Table 171. 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 "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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.
177	245	631	Object Type	Char(8)	The type of object.
		639	ASP Name	Char(10)	ASP name for CRQD library

*Table 171. CQ (*CRQD Changes) journal entries.* QASYCQJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		649	ASP Number	Char(5)	ASP number for CRQD library

CU (Cluster Operations) journal entries

This table provides the format of the CU (Cluster Operations) journal entries.

Table 172. CU (Cluster Operations) journal entries. QASYCUJ4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630 and "Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632 for field listing.
	224	610	Entry Type	Char(1)	The type of entry. **M** Cluster control operation **R** Cluster Resource Group (*CRG) management operation **P** Cluster policy operation

Table 172. CU (Cluster Operations) journal entries. QASYCUJ4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	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 **RCY** CHGCLURCY command **RMV** Remove **RSC** Report state change **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. This field will only contain data when entry type (J5 offset 610) is R.
	241	627	CRG Library Name	Char(10)	The Cluster Resource Group object library. This field will only contain data when entry type (J5 offset 610) is R.
	251	637	Cluster Name	Char(10)	The name of the cluster.

Table 172. CU (Cluster Operations) journal entries. QASYCUJ4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	261	647	Node ID	Char(8)	The node ID. This field will only contain data when entry type (J5 offset 610) is M or R.
	269	655	Source Node ID	Char(8)	The source node ID. This field will only contain data when entry type (J5 offset 610) is M or R.
	277	663	Source User Name	Char(10)	Name of the source system user that initiated the request. This field will only contain data when entry type (J5 offset 610) is M or R.
	287	673	User Queue Name	Char(10)	Name of the user queue where responses are sent. This field will only contain data when entry type (J5 offset 610) is M or R.
	297	683	User Queue Library	Char(10)	The user queue library. This field will only contain data when entry type (J5 offset 610) is M or R.
		693	ASP Name	Char(10)	ASP name for user queue library. This field will only contain data when entry type (J5 offset 610) is M or R.
		703	ASP Number	Char(5)	ASP number for user queue library. This field will only contain data when entry type (J5 offset 610) is M or R.
		708	Policy Name	Char(32)	Cluster policy name. This field will only contain data when entry type (J5 offset 610) is P.
		740	Application ID	Char(20)	Application identifier. This field will only contain data when entry type (J5 offset 610) is P.
		760	Domain Type	Char(10)	Domain type. This field will only contain data when entry type (J5 offset 610) is P. **ADMDMN** Cluster administrative domain. **CRG** Cluster resource group
		770	Domain Name	Char(10)	Domain name. This field will only contain data when entry type (J5 offset 610) is P.
		780	Policy Qualifier	Char(64)	Policy qualifier. This field will only contain data when entry type (J5 offset 610) is P.

CV (Connection Verification) journal entries

This table provides the format of the CV (Connection Verification) journal entries.

Table 173. CV (Connection Verification) journal entries. QASYCVJ4/J5 Field Description File

Offset			Field	Format	Description
JE	**J4**	**J5**			
	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630 and "Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632 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 173. CV (Connection Verification) journal entries. QASYCVJ4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	236	622	Protocol	Char(10)	The type of entry. **L2TP** Layer Two Tunneling protocol **PPP** Point-to-Point protocol. **SLIP** Serial Line Internet Protocol.
	246	632	Local Authenticatio n Method	Char(10)	The type of entry. **CHAP** Challenge Handshake Authentication Protocol. **PAP** Password Authentication Protocol. **SCRIPT** Script method.
	256	642	Remote Authenticatio n 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 173. CV (Connection Verification) journal entries. QASYCVJ4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
	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 Session 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	Reserved	Binary(4)	Always zero
		1025	ASP Name	Char(10)	ASP name for validation list library
		1035	ASP Number	Char(5)	ASP number for validation list library

CY (Cryptographic Configuration) journal entries

This table provides the format of the CY (Cryptographic Configuration) journal entries.

Table 174. CY (Cryptographic Configuration) journal entries. QASYCYJ4/J5 Field Description File

Offset			Field	Format	Description
JE	**J4**	**J5**			
	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 for field listing.
	224	610	Entry Type	Char(1)	The type of entry. **A** Cryptographic Coprocessor Access Control Function **F** Cryptographic Coprocessor Facility Control Function **K** Cryptographic Services Master Key Function **M** Cryptographic Coprocessor Master Key Function

Table 174. CY (Cryptographic Configuration) journal entries. QASYCYJ4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
	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 or clear FCV. **INI** Reinitialize card. **LOD** Load master key. **QRY** Query role or profile information. **RCP** Replace a card profile. **RCR** Replace a card role. **RCV** Receive master keys. **SDL** Set dual control flow mode. **SET** Set master keys. **SHR** Cloning shares. **SSG** Set single control flow mode. **TST** Test master key.

Table 174. CY (Cryptographic Configuration) journal entries. QASYCYJ4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
	228	614	Card Profile	Char(8)	The name of the card profile.[2]
	236	622	Card Role	Char(8)	The role of the card profile.[2]
	244	630	Device Name	Char(10)	The cryptographic device name or the resource name.[2]
		640	Master Key ID[1]	Binary(4)	The cryptographic services Master Key ID[3]. Possible values are as follows: **-2** Save/restore master key **-1** ASP master key **1** Master key 1 **2** Master key 2 **3** Master key 3 **4** Master key 4 **5** Master key 5 **6** Master key 6 **7** Master key 7 **8** Master key 8
		644	Master key encryption	Char(1)	Master Key encrypted with default S/R Master Key. **Y** The master key was set and encrypted with the default Save/Restore Master Key. **N** The master key was set and encrypted with a user-set Save/Restore Master Key.
		645	Master key version	Char(8)	The version of the master key that was cleared. **NEW** The new version was cleared. **CURRENT** The current version was cleared. **OLD** The old version was cleared. **PENDING** The pending version was cleared.

Table 174. CY (Cryptographic Configuration) journal entries. QASYCYJ4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1 When the entry type (J5 offset 610) is K, the card profile (J5 offset 614), card role (J5 offset 622), and device name (J5 offset 630) is set to blanks. **2** When the entry type is K, this field is blank. **3** When the entry type is not K, this field is blank.					

DI (Directory Server) journal entries

This table provides the format of the DI (Directory Server) journal entries.

Table 175. DI (Directory Server) journal entries. QASYDIJ4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 for field listing.
	224	610	Entry Type	Char(1)	The type of entry. **L** LDAP Operation

Table 175. DI (Directory Server) journal entries. QASYDIJ4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	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. **CI** Create instance **CO** Object creation. **CP** Password change. **DI** Delete instance **DO** Object delete. **EX** LDAP directory export. **IM** LDAP directory import. **OM** Object management (rename). **OW** Ownership change. **PO** Policy change. **PW** Password fail. **RM** Replication management **UB** Successful unbind. **ZC** Object change. **ZR** Object read.

Table 175. DI (Directory Server) journal entries. QASYDIJ4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
	227	613	Authority Failure Code	Char(1)	Code for authority failures. This field is used only if the operation type (J5 offset 611) 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, publishing). **G** Unauthorized replication management attempt. **I** Unauthorized import attempt. **M** Unauthorized change attempt. **P** Unauthorized policy change attempt. **R** Unauthorized read (search) attempt. **U** Unauthorized attempt to read the audit configuration. **X** Unauthorized proxy authorization attempt.

Table 175. DI (Directory Server) journal entries. QASYDIJ4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	228	614	Configuration Change	Char(1)	Configuration changes. This field is only used if the operation type (J5 offset 611) is CF, RM, CA or OW. If the operation type (J5 offset 611) is CF this field will contain: **A** Administrator ND change. **C** Change log on or off. **L** Backend library name change. **P** Publishing agent change. **R** Replica server change. If the operation type (J5 offset 611) is RM this field will contain: **U** Suspend replication. **V** Resume replication. **W** Replicate pending changes now. **X** Skip one or more pending changes. **Y** Quiesce replication context. **Z** Unquiesce replication context. If the operation type (J5 offset 611) is CA or OW this field will contain the previous setting of the owner or ACL propagate value. **T** True **F** False
	229	615	Configuration Change Code	Char(1)	Code for configuration changes. This field is used only if the operation type (J5 offset 611) is CF. **A** Item added to configuration **D** Item deleted from configuration **M** Item modified

Table 175. DI (Directory Server) journal entries. QASYDIJ4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	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 (J5 offset 611) is CA or OW. **T** True **F** False
	231	617	Bind Authenticatio n Choice	Char(20)	The bind authentication choice. This field is used only if the operation type (J5 offset 611) 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 TLS was used on the request. This field is used only 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.

Table 175. DI (Directory Server) journal entries. QASYDIJ4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	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	Name CCSID	Bin(5)	The coded character set identifier of the name. This field is used only if the operation type (J5 offset 611) is OW or AD. • For operation type OW, this field will contain the CCSID of the previous owner name. • For operation type AD, this field will contain the CCSID of the previous audit value.
	4347	4733	Name Length	Bin(4)	The length of the name. This field is used only if the operation type is OW or AD. • For operation type OW, this field will contain the length of the previous owner name. • For operation type AD, this field will contain the length of the previous audit value.
	4349	4735	Name[1]	Char(2002)	The name. This field is used only if the operation type (J5 offset 611) is OW or AD. • For operation type OW, this field will contain the previous owner name. • For operation type AD, this field will contain the previous audit value.
	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 (J5 offset 611) is OM, OW, PO, ZC, AF+M, or AF+P. • 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 PO, ZC, AF+M, or AF+P, this field will contain the CCSID of the list of changed attribute types in the New Name field.

Table 175. DI (Directory Server) journal entries. QASYDIJ4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	6355	6741	New Name Length	Bin(4)	The length of the new name. This field is used only if the operation type (J5 offset 611) is OM, OW, PO, ZC, AF+M, or AF+P. • 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 PO, ZC, AF+M,or AF+P, 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 (J5 offset 611) is OM, OW, PO, ZC, AF+M, or AF+P. • 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 PO, ZC, AF+M, or AF+P, 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 path name.
	8394	8780	Path Name Country or Region ID[2]	Char(2)	The Country or Region ID of the path name.
	8396	8782	Path Name Language ID[2]	Char(3)	The language ID of the path name.
	8399	8785	Path Name Length[2]	Bin(4)	The length of the path name.

Table 175. DI (Directory Server) journal entries. QASYDIJ4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	8401	8787	Path Name Indicator[2]	Char(1)	Path name indicator. **Y** The Path Name field contains complete absolute path name for the object. **N** The Path Name field does not contain an absolute path name for the object, instead it contains a relative path name. The Relative Directory File ID field is valid and can be used to form an absolute path name with this relative path name.
	8402	8788	Relative Directory File ID[2,3]	Char(16)	When the Path Name Indicator field is N, this field contains the file ID of the directory that contains the object identified in the Path Name field. Otherwise it contains hex zeros.[3]
	8418	8804	Path Name[1,2]	Char(5002)	The path name of the object.
		13806	Local User Profile	Char(10)	The local user profile name that is mapped to the LDAP user name (J5 offset 719). Blank indicates no user profile is mapped.
		13816	Administrator Indicator	Char(1)	Administrator indicator for the LDAP user name (J5 offset 719). **Y** The LDAP user is an administrator. **N** The LDAP user is not an administrator. **U** It is unknown at this time if the LDAP user is an administrator.
		13817	Proxy ID CCSID	Bin(5)	The coded character set identifier (CCSID) of the proxy ID.
		13821	Proxy ID Length	Bin(4)	The length of the proxy ID.
		13823	Proxy ID[1]	Char(2002)	The name of the proxy ID. This field is used when the proxy authorization control is used to request that an operation be done under the authority of the proxy ID, or for a SASL bind in which the client has specified an authorization ID different from the bind ID.
		15825	Group Assertion	Char(1)	Group membership assertion **0** Groups were not specified by client. **1** Groups were specified by client.

Table 175. DI (Directory Server) journal entries. QASYDIJ4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		15826	Cross Reference	Char(36)	Cross reference string used to correlate this entry with the XD entry/entries listing the groups.
		15862	Instance Name	Char(8)	Instance name
		15870	Route CCSID	Bin(5)	CCSID of route
		15874	Route Length	Bin(4)	Length of route
		15876	Route	Char(502)	Request route

1

This is a variable length field. The first two bytes contain the length of the value in the field.

2

These fields are used only if the operation type (J5 offset 611) is EX or IM.

3

If the Path Name Indicator field is N, but the Relative Directory File ID is hex zeros, then there was some error in determining the path name information.

DO (Delete Operation) journal entries

This table provides the format of the DO (Delete Operation) journal entries. Objects deleted from QTEMP library are not audited.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_DO table function: AUDIT_JOURNAL_DO

Table 176. DO (Delete Operation) journal entries. QASYDOJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See “Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)” on page 630,“Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)” on page 632, and “Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)” on page 633 for field listing.

Table 176. DO (Delete Operation) journal entries. QASYDOJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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 where the object is stored.
177	245	631	Object Type	Char(8)	The type of object.
185	253		(Reserved Area)	Char(20)	
		639	Object Attribute	Char(10)	The attribute of the object.
		649	(Reserved Area)	Char(10)	
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.
332	400	786	Object Name Country or Region ID[1]	Char(2)	The Country or Region ID for the object name.

Table 176. DO (Delete Operation) journal entries. QASYDOJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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 path name.
	987	1373	Path Name Country or Region ID	Char(2)	The Country or Region ID for the path name.
	989	1375	Path Name Language ID	Char(3)	The language ID for the path name.
	992	1378	Path Name Length	Binary(4)	The length of the path name.
	994	1380	Path Name Indicator	Char(1)	Path name indicator: **Y** The Path Name field contains complete absolute path name for the object. **N** The Path Name field does not contain an absolute path name for the object, instead it contains a relative path name. The Relative Directory File ID field is valid and can be used to form an absolute path name with this relative path name.
	995	1381	Relative Directory File ID[3]	Char(16)	When the Path Name Indicator field is N, this field contains the file ID of the directory that contains the object identified in the Path Name field. Otherwise it contains hex zeros.[3]
	1011	1397	Path Name[4]	Char(5002)	The path name of the object.

Table 176. DO (Delete Operation) journal entries. QASYDOJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1 These fields are used only for objects in the "root" (/), QOpenSys, 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** If the Path Name Indicator field is N, but the Relative Directory File ID is hex zeros, then there was some error in determining the path name information. **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.					

DS (Service Tools User ID and Attribute Changes) journal entries

This table provides the format of the DS (Service Tools User ID and Attribute Changes) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_DS table function: AUDIT_JOURNAL_DS

Table 177. DS (Service Tools User ID and Attribute Changes) journal entries. QASYDSJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See “Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)” on page 630,“Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)” on page 632, and “Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)” on page 633 for field listing.

Table 177. DS (Service Tools User ID and Attribute Changes) journal entries. QASYDSJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
156	224	610	Entry Type	Char(1)	The type of entry. **A** Reset of a service tools user ID password using the CHGDSTPWD command. **C** Change to a service tools user ID using the QSYCHGDS API. **D** Delete of a service tools user ID using the DLTSSTUSR command. **H** Change to a service tools user ID using the CHGSSTUSR command. **P** Change to a service tools user ID password using the QSYCHGDS API. **R** Create of a service tools user ID using the CRTSSTUSR command. **S** Change to the service tools security attributes using the CHGSSTSECA command.
157	225	611	IBM-Supplied Service Tools User ID Reset	Char(1)	**Y** Request to reset an IBM-supplied service tools user ID. This field only contains data when Entry type (J5 offset 610) is A.
158	226	612	Service Tools User ID to change	Char(10)	The service tools user ID to change. This field only contains data when Entry type (J5 offset 610) is C or P. It may contain one of the following special values. ***SECURITY** ***FULL** ***BASIC**
168	236	622	Service Tools User ID New Name	Char(8)	The new name of the service tools user ID. This field only contains data when Entry type (J5 offset 610) is C and the new service tools user ID name length is 8 bytes or less.
176	244	630	Service Tools User ID Password Change	Char(1)	Request to change the service tools user ID password. This field only contains data when Entry type (J5 offset 610) is P. **Y** Request to change service tools user ID password.

Table 177. DS (Service Tools User ID and Attribute Changes) journal entries. QASYDSJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
	245	631	Service Tools User ID	Char(10)	When Entry type (J5 offset 610) is C this field contains the new name of the service tools user ID. When Entry type is D, H, or R this field contains the service tools user ID being created, changed, or deleted.
	255	641	Service Tools User ID Requesting Profile	Char(10)	The name of the service tools user ID that requested the action. This field only contains data when Entry type (J5 offset 610) is C, D, H, P, R, or S.
		651	Status	Char(10)	Status of the user ID. This field only contains data when Entry type (J5 offset 610) is H or R. ***ENABLED** ***DISABLED**
		661	Previous Status	Char(10)	Previous status of the user ID. This field only contains data when Entry type (J5 offset 610) is H. ***ENABLED** ***DISABLED**
		671	Set Password Expired	Char(1)	Set password to expired. This field only contains data when Entry type (J5 offset 610) is H or R. **Y** Password is expired
		672	Linked Profile	Char(10)	The user profile that is linked to the service tools user ID. This field only contains data when Entry type (J5 offset 610) is H or R.
		682	Previous Linked Profile	Char(10)	The user profile that was previously linked to the service tools user ID. This field only contains data when Entry type (J5 offset 610) is H.
		692	(Reserved Area)	Char(10)	
		Current Privileges - The privilege fields only contain data when Entry type (J5 offset 610) is H or R. **Y** Service tools user ID has the privilege **N** Service tools user ID does not have the privilege			
		702		Char(1)	Disk units - operations
		703		Char(1)	Disk units - administration
		704		Char(1)	Disk units - read only

Table 177. DS (Service Tools User ID and Attribute Changes) journal entries. QASYDSJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		705		Char(1)	System partitions - operations
		706		Char(1)	System partitions - administration
		707		Char(1)	Partition remote panel key
		708		Char(1)	Operator panel functions
		709		Char(1)	Operating system initial program load (IPL)
		710		Char(1)	Install
		711		Char(1)	Performance data collector
		712/		Char(1)	Hardware service manager
		713		Char(1)	Display/Alter/Dump
		714		Char(1)	Main storage dump
		715		Char(1)	Product activity log
		716		Char(1)	Licensed Internal Code log
		717		Char(1)	Licensed Internal Code fixes
		718		Char(1)	Trace
		719		Char(1)	Dedicated Service Tools (DST) environment
		720		Char(1)	Remote service support
		721		Char(1)	Service tools security
		722		Char(1)	Service tools save and restore
		723		Char(1)	Debug
		724		Char(1)	System capacity - operations
		725		Char(1)	System capacity - administrator
		726		Char(1)	System security
		727		Char(1)	Start service tools
		728		Char(1)	Take over console
		729	(Reserved Area)	Char(13)	
		Previous Privileges - The privilege fields only contain data when Entry type (J5 offset 610) is H. **Y** Service tools user ID has the privilege **N** Service tools user ID does not have the privilege			
		742		Char(1)	Disk units - operations
		743		Char(1)	Disk units - administration

Table 177. DS (Service Tools User ID and Attribute Changes) journal entries. QASYDSJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		744		Char(1)	Disk units - read only
		745		Char(1)	System partitions - operations
		746		Char(1)	System partitions - administration
		747		Char(1)	Partition remote panel key
		748		Char(1)	Operator panel functions
		749		Char(1)	Operating system initial program load (IPL)
		750		Char(1)	Install
		751		Char(1)	Performance data collector
		752		Char(1)	Hardware service manager
		753		Char(1)	Display/Alter/Dump
		754		Char(1)	Main storage dump
		755		Char(1)	Product activity log
		756		Char(1)	Licensed Internal Code log
		757		Char(1)	Licensed Internal Code fixes
		758		Char(1)	Trace
		759		Char(1)	Dedicated Service Tools (DST) environment
		760		Char(1)	Remote service support
		761		Char(1)	Service tools security
		762		Char(1)	Service tools save and restore
		763		Char(1)	Debug
		764		Char(1)	System capacity - operations
		765		Char(1)	System capacity - administrator
		766		Char(1)	System security
		767		Char(1)	Start service tools
		768		Char(1)	Take over console
		769	(Reserved Area)	Char(13)	
		782	SST Password Level	Char(1)	System Service Tools (SST) password level. This field only contains data when Entry Type (J5 offset 610) is S.
		783	Previous SST Password Level	Char(1)	Previous (SST) password level. This field only contains data when Entry Type (J5 offset 610) is S.

Table 177. DS (Service Tools User ID and Attribute Changes) journal entries. QASYDSJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		784	Allow System Value Changes	Char(1)	Allow changes to security related system values. This field only contains data when Entry Type (J5 offset 610) is S. **Y** Allow changes
		785	Previous Allow System Value Changes	Char(1)	Previous value of allow changes to security related system values. This field only contains data when Entry Type (J5 offset 610) is S. **Y** Allow changes
		786	(Reserved Area)	Char(10)	
		Current Password Rules - These fields only contain data when Entry Type (J5 offset 610) is S.			
		796	Limit Profile Name	Char(1)	Limit profile name. **Y** The password may not contain the upper case profile name.
		797	Hours to Block	Char(6)	The number of hours during which the password is blocked from being changed. ***NONE** There is no restriction on how frequently a user can change a password.
		803	Minimum Password Length	Char(6)	Minimum password length.
		809	Maximum Password Length	Char(6)	Maximum password length.
		815	Use From 3 Groups	Char(1)	The password must contain characters from at least three of the four types of characters: Uppercase letters, lowercase letters, digits, and special characters. **Y** The password must contain characters from at least three of the four groups.
		816	Limit Adjacent Characters	Char(1)	Limit adjacent characters. **Y** The password may not contain two or more adjacent characters.

Table 177. DS (Service Tools User ID and Attribute Changes) journal entries. QASYDSJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		817	Limit Repeating Characters	Char(1)	Limit repeating characters. **Y** The password may not contain two or more occurrences of the same character.
		818	Limit Same Position	Char(1)	Limit characters in the same position. **Y** The same character may not be used in the same position as in the previous password.
		819	Minimum Digits	Char(6)	The minimum number of digit characters that must occur in the password. ***NONE** No digits are required.
		825	Maximum Digits	Char(6)	The maximum number of digit characters that may occur in the password. ***NOMAX** Any number of digits are allowed in the password.
		831	Limit Adjacent Digits	Char(1)	Limit adjacent digits. **Y** The password must not contain two or more adjacent (consecutive) digits.
		832	Limit Digit First	Char(1)	Limit digit in first position. **Y** The first character of the password must not be a digit.
		833	Limit Digit Last	Char(1)	Limit digit in last position. **Y** The last character of the password must not be a digit.
		834	Minimum Letters	Char(6)	The minimum number of letter characters that must occur in the password. ***NONE** No letters are required.
		840	Maximum Letters	Char(6)	The maximum number of letter characters that may occur in the password. ***NOMAX** Any number of letters are allowed in a password.

Table 177. DS (Service Tools User ID and Attribute Changes) journal entries. QASYDSJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		846	Limit Adjacent Letters	Char(1)	Limit adjacent letters. **Y** The password must not contain two or more adjacent (consecutive) letters.
		847	Limit Letter First	Char(1)	Limit letter in first position. **Y** The first character of the password must not be a letter.
		848	Limit Letter Last	Char(1)	Limit letter in last position. **Y** The last character of the password must not be a letter.
		849	Number Mixed Case Letters	Char(6)	The password must contain at least the specified number of uppercase and lowercase letters. ***NONE** Mixed case letters are not required in a password.
		855	Minimum Special Characters	Char(6)	The minimum number of special characters that must occur in the password. ***NONE** No special characters are required.
		861	Maximum Special Characters	Char(6)	The maximum number of special characters that may occur in the password. ***NOMAX** Any number of special characters are allowed in a password.
		867	Limit Adjacent Special Characters	Char(1)	Limit adjacent special characters. **Y** The password must not contain two or more adjacent (consecutive) special characters.
		868	Limit Special Character First	Char(1)	Limit special character in first position. **Y** The first character of the password must not be a special character.
		869	Limit Special Character Last	Char(1)	Limit special character in last position. **Y** The last character of the password must not be a special character.
		870	(Reserved Area)	Char(10)	

Table 177. DS (Service Tools User ID and Attribute Changes) journal entries. QASYDSJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		Previous Password Rules - These fields only contain data when Entry Type (J5 offset 610) is S.			
		880	Previous Limit Profile Name	Char(1)	Limit profile name. **Y** The password may not contain the upper case profile name.
		881	Previous Hours to Block	Char(6)	The number of hours during which the password is blocked from being changed. ***NONE** There is no restriction on how frequently a user can change a password.
		887	Previous Minimum Password Length	Char(6)	Minimum password length.
		893	Previous Maximum Password Length	Char(6)	Maximum password length.
		899	Previous Use From 3 Groups	Char(1)	The password must contain characters from at least three of the four types of characters: Uppercase letters, lowercase letters, digits, and special characters. **Y** The password must contain characters from at least three of the four groups.
		900	Previous Limit Adjacent Characters	Char(1)	Limit adjacent characters. **Y** The password may not contain two or more adjacent characters.
		901	Previous Limit Repeating Characters	Char(1)	Limit repeating characters. **Y** The password may not contain two or more occurrences of the same character.
		902	Previous Limit Same Position	Char(1)	Limit characters in the same position. **Y** The same character may not be used in the same position as in the previous password.

Table 177. DS (Service Tools User ID and Attribute Changes) journal entries. QASYDSJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		903	Previous Minimum Digits	Char(6)	The minimum number of digit characters that must occur in the password. ***NONE** No digits are required.
		909	Previous Maximum Digits	Char(6)	The maximum number of digit characters that may occur in the password. ***NOMAX** Any number of digits are allowed in the password.
		915	Previous Limit Adjacent Digits	Char(1)	Limit adjacent digits. **Y** The password must not contain two or more adjacent (consecutive) digits.
		916	Previous Limit Digit First	Char(1)	Limit digit in first position. **Y** The first character of the password must not be a digit.
		917	Previous Limit Digit Last	Char(1)	Limit digit in last position. **Y** The last character of the password must not be a digit.
		918	Previous Minimum Letters	Char(6)	The minimum number of letter characters that must occur in the password. ***NONE** No letters are required.
		924	Previous Maximum Letters	Char(6)	The maximum number of letter characters that may occur in the password. ***NOMAX** Any number of letters are allowed in a password.
		930	Previous Limit Adjacent Letters	Char(1)	Limit adjacent letters. **Y** The password must not contain two or more adjacent (consecutive) letters.
		931	Previous Limit Letter First	Char(1)	Limit letter in first position. **Y** The first character of the password must not be a letter.

Table 177. DS (Service Tools User ID and Attribute Changes) journal entries. QASYDSJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		932	Previous Limit Letter Last	Char(1)	Limit letter in last position. **Y** The last character of the password must not be a letter.
		933	Previous Number Mixed Case Letters	Char(6)	The password must contain at least the specified number of uppercase and lowercase letters. ***NONE** Mixed case letters are not required in a password.
		939	Previous Minimum Special Characters	Char(6)	The minimum number of special characters that must occur in the password. ***NONE** No special characters are required.
		945	Previous Maximum Special Characters	Char(6)	The maximum number of special characters that may occur in the password. ***NOMAX** Any number of special characters are allowed in a password.
		951	Previous Limit Adjacent Special Characters	Char(1)	Limit adjacent special characters. **Y** The password must not contain two or more adjacent (consecutive) special characters.
		952	Previous Limit Special Character First	Char(1)	Limit special character in first position. **Y** The first character of the password must not be a special character.
		953	Previous Limit Special Character Last	Char(1)	Limit special character in last position. **Y** The last character of the password must not be a special character.

EV (Environment Variable) journal entries

This table provides the format of the EV (Environment Variable) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_EV table function: AUDIT_JOURNAL_EV

Table 178. EV (Environment Variable) journal entries. QASYEVJ4/J5 Field Description File

Offset			Field	Format	Description
JE	**J4**	**J5**			
	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 for field listing.
	224	610	Entry Type	Char(1)	The type of entry. **A** Add **C** Change **D** Delete **I** Initialize Environment Variable Space
	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 Value Truncated[1]	Char(1)	Indicates whether the new environment variable value (offset 1241) is truncated. **Y** Environment variable value truncated. **N** Environment variable value not truncated.
	1235	1621	New Value CCSID[1]	Binary(5)	The CCSID of the new environment variable value.
	1239	1625	New Value Length[1]	Binary(4)	The length of the new environment variable value.
	1241	1627	New Environment Variable Value[1,2]	Char (1002)	The new environment variable value.

Table 178. EV (Environment Variable) journal entries. QASYEVJ4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1 These fields are used when the entry type is A or C. **2** This is a variable length field. The first two bytes contain the length of the environment variable name.					

GR (Generic Record) journal entries

This table provides the format of the GR (Generic Record) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_GR table function: AUDIT_JOURNAL_GR

Table 179. GR (Generic Record) journal entries. QASYGRJ4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	1	1			Heading fields common to all entry types. See “Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)” on page 630 and “Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)” on page 632 for field listing.
	224	610	Entry Type	Char(1)	The type of entry. **A** Exit program added **C** Operations Resource Monitoring and Control Operations **D** Exit program removed **F** Function registration operations **O** ObjectConnect operations **R** Exit program replaced

Table 179. GR (Generic Record) journal entries. QASYGRJ4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
	225	611	Action	Char(2)	The action performed. **ZC** Change **ZR** Read For entry type O, the possible values are: **SV** Save **RS** Restore
	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. For entry type O, this field contains the name of the user performing the ObjectConnect operation.
	237	623	Field 1 CCSID	Binary (5)	The CCSID value for field 1.
	241	627	Field 1 Length	Binary (4)	The length of the data in field 1.

Table 179. GR (Generic Record) journal entries. QASYGRJ4/J5 Field Description File *(continued)*					
Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	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 been 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. For entry type C, this field contains the name of the RMC function that is being attempted. The possible values are: • **mc_reg_event_select** Register event using attribute selection • **mc_reg_event_handle** Register event using resource handle • **mc_reg_class_event** Register event for a resource class • **mc_unreg_event** Unregister event • **mc_define_resource** Define new resource • **mc_undefine_resource** Undefine resource • **mc_set_select** Set resource attribute values using attribute selection • **mc_set_handle** Set resource attribute values using resource handle • **mc_class_set** Set resource class attribute values • **mc_query_p_select** Query resource persistent attributes using attribute selection • **mc_query_d_select** Query resource dynamic attributes using attribute selection

Table 179. GR (Generic Record) journal entries. QASYGRJ4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	J4	J5			
					• **mc_query_p_handle** Query resource persistent attributes using resource handle • **mc_query_d_handle** Query resource dynamic attributes using resource handle • **mc_class_query_p** Query resource class persistent attributes • **mc_class_query_d** Query resource class dynamic attributes • **mc_qdef_resource_class** Query resource class definition • **mc_qdef_p_attribute** Query persistent attribute definition • **mc_qdef_d_attribute** Query dynamic attribute definition • **mc_qdef_sd** Query Structured Data definition • **mc_qdef_valid_values** Query definition of a persistent attribute's valid values • **mc_qdef_actions** Query definition of a resource's actions • **mc_invoke_action** Invoke action on a resource • **mc_invoke_class_action** Invoke action on a resource class For entry type O, this field contains the ObjectConnect CL command. The possible values are: **SAVRST** Save/Restore Integrated File System **SAVRSTCFG** Save/Restore Configuration **SAVRSTCHG** Save/Restore Changed Object **SAVRSTDLO** Save/Restore Document Library Object **SAVRSTLIB** Save/Restore Library **SAVRSTOBJ** Save/Restore Object
	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.

Table 179. GR (Generic Record) journal entries. QASYGRJ4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	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. For entry type C, this field contains the name of the resource or resource class against which the operation was attempted. For entry type O and Action RS (J5 offset 611), this field contains the name of the system on which the objects are saved. For entry type O and Action SV (J5 offset 611), this field contains the name of the system on which the objects are restored.
	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 179. GR (Generic Record) journal entries. QASYGRJ4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	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 values: ***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. For entry type C, this field contains the result of any authorization check that was made for the operation indicated in field 1. The following are possible values: • *NOAUTHORITYCHECKED: When either the operation indicated in field 1 does not require an authorization check, or if for any other reason an authorization check was not attempted. • *AUTHORITYPASSED: When the mapped user ID indicated in the User Profile Name has successfully passed the appropriate authorization check for the operation indicated in field 1 against the resource or resource class indicated in field 2. • *AUTHORITYFAILED: When the mapped user ID indicated in the User Profile Name has failed the appropriate authorization check for the operation indicated in field 1 against the resource or resource class indicated in field 2.

Table 179. GR (Generic Record) journal entries. QASYGRJ4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
					For entry type O and Commands (J5 offset 629) SAVRST, SAVRSTCHG, SAVRSTLIB, and SAVRSTOBJ, this field contains two pieces of information used for the save. The information is in the following order: • Char(10) Library. The library name is set when processing objects from the QSYS file system otherwise it is blank. It is the name of the saved library or the library from which the objects were saved. • Char(10) ASP device or ASP number. The value may be blank or one of the following: – name – * – *ALLAVL – *CURASPGRP – *SYSBAS – *ANY – 1-32
	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.

Table 179. GR (Generic Record) journal entries. QASYGRJ4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
	567	953	Field 4	Char(102)[1]	Field 4 data. For entry type F (J5 offset 610), there is a value for this field only if the function registration operation is one of the following values: ***CHGUSAGE** When the operation is *CHGUSAGE, this field contains the previous usage value for a user. ***REGISTER** When the operation is *REGISTER, this field contains the allow *ALLOBJ setting for the function. ***REREGISTER** When the operation is *REREGISTER, this field contains the allow *ALLOBJ setting for the function. For entry type O and Commands (J5 offset 629) SAVRST, SAVRSTCHG, SAVRSTLIB, and SAVRSTOBJ, this field contains three pieces of information used for the restore. The information is in following order: • Char(10) Library name. The library name is set when processing objects from the QSYS file system otherwise it is blank. It is the name of the restored library or the library to which the objects were restored. • Char(10) ASP device. The ASP device is one of the following or blank if ASP number is set: – name – *SAVASPDEV • Char(10) ASP number. The ASP number is one of the following or blank if ASP device is set: – 1-32 – *SAVASP
		1055	Field 5 CCSID	Binary (5)	The CCSID value for field 5.
		1059	Field 5 Length	Binary (4)	The length of the data in field 5.

Table 179. GR (Generic Record) journal entries. QASYGRJ4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		1061	Field 5	Char(102)[1]	Field 5 data. For entry type F (J5 offset 610), this field contains the previous default usage value. There is a value for this field only if the function registration operation (J5 offset 629) is *REREGISTER. The user name (J5 offset 613) will be *DEFAULT. For entry type O, this field contains the UUID of the ObjectConnect operation.
		1163	Field 6 CCSID	Binary (5)	The CCSID value for field 6.
		1167	Field 6 Length	Binary (4)	The length of the data in field 6.
		1169	Field 6	Char(102)[1]	Field 6 data. For entry type F (J5 offset 610), this field contains the previous allow *ALLOBJ setting for the function. There is a value for this field only if the function registration operation (J5 offset 629) is *REREGISTER. For entry type O and Action SV (J5 offset 611), this field contains the name of the user under which the restore will be performed. The possible values are: • name • *NONE • *CURRENT • *KERBEROS

1
This is a variable length field. The first two bytes contain the length of the field.

GS (Give Descriptor) journal entries

This table provides the format of the GS (Give Descriptor) journal entries.

Table 180. GS (Give Descriptor) journal entries. QASYGSJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 for field listing.

Table 180. GS (Give Descriptor) journal entries. QASYGSJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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.
183	251	637	User Profile Name	Char (10)	The name of the user profile.
	261	647	JUID	Char (10)	The Job User ID of the target job. (This value applies only to subtype G audit records.)

IM (Intrusion Monitor) journal entries

This table provides the format of the IM (Intrusion Monitor) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_IM table function: AUDIT_JOURNAL_IM

Table 181. IM (Intrusion Monitor) journal entries. QASYIMJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		1			Heading fields common to all entry types.
		610	Entry Type	Char(1)	The type of entry. **P** Potential intrusion event detected
		611	Time of Event	TIMESTAMP	The time that the event was detected, in SAA timestamp format.
		637	Detection Point Identifier	Char(4)	A unique identifier for the processing location that detected the intrusion event. This field is intended for use by service personnel.
		641	Local Address Family	Char(1)	Local IP address family associated with the detected event.
		642	Local Port Number	Zone(5, 0)	Local port number associated with the detected event.
		647	Local IP Address	Char(46)	Local IP address associated with the detected event.

Table 181. IM (Intrusion Monitor) journal entries. QASYIMJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		693	Remote Address Family	Char(1)	Remote address family associated with the detected event.
		694	Remote Port Number	Zoned(5, 0)	Remote port number associated with the detected event.
		699	Remote IP Address	Char(46)	Remote IP address associated with the detected event.
		745	Probe Type Identifier	Char(6)	Identifies the type of probe used to detect the potential intrusion. Possible values are as follows: **ATTACK** Attack action detected event **TR-TCP** Traffic Regulation action detected event over TCP **TR-SSL** Traffic Regulation action detected System TLS failed handshake event **TR-UDP** Traffic Regulation action detected event over UDP **SCANE** Scan event action detected event **SCANG** Scan global action detected event **XATTAC** Possible extrusion attack **XTRTCP** Outbound TR detected event (TCP) **XTRUDP** Outbound TR detected event (UDP) **XSCAN** Outbound scan event detected
		751	Event Correlator	Char(4)	Unique identifier for this specific intrusion event. This identifier can be used to correlate this audit record with other intrusion detection information.

Table 181. IM (Intrusion Monitor) journal entries. QASYIMJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		755	Event type	Char(8)	Identifies the type of potential intrusion that was detected. The possible values are as follows: **ACKSTORM** TCP ACK storm **ADRPOISN** Address poisoning **FLOOD** Flood event **FRAGGLE** Fraggle attack **ICMPRED** ICMP (Internet Control Message Protocol) redirect **IPFRAG** IP fragment **MALFPKT** Malformed packet **OUTRAW** Outbound Raw **PERPECH** Perpetual echo **PNGDEATH** Ping of death **RESTOPT** Restricted IP options **RESTPROT** Restricted IP protocol **SMURF** Smurf attack
		763	Protocol	Char(3)	Protocol number
		766	Condition	Char(4)	Condition number from IDS policy file
		770	Throttling	Char(1)	• 0 = not active • 1 = active
		771	Discarded Packets	Zoned(5,0)	Number of discarded packets when throttled
		776	Target TCP/IP Stack	Char(1)	**P** Production Stack **S** Service Stack
		777	Reserved	Char(6)	Reserved for future use

Table 181. IM (Intrusion Monitor) journal entries. QASYIMJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		783	Suspected Packet	Char(1002)[1]	A variable length field which can contain up to the first 1000 bytes of the IP packet associated with the detected event. This field contains binary data and should be treated as if it has a CCSID of 65535. When Probe Type Identifier (offset 745) is 'TR-SSL', this field contains a blank padded character string that indicates error information for the failing handshake. The first 2 bytes of this field contain the length of the error information. Following the length is a 6-byte character string that represents the processing location that detected the failed handshake. Following the 6-byte string is a 40-byte character string that indicates the error code that is returned on the failing handshake.

1
This is a variable length field. The first 2 bytes contain the length of the suspected packet information.

IP (Interprocess Communication) journal entries

This table provides the format of the IP (Interprocess Communication) journal entries.

Table 182. IP (Interprocess Communication) journal entries. QASYIPJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See “Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)” on page 630,“Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)” on page 632, and “Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)” on page 633 for field listing.

Table 182. IP (Interprocess Communication) journal entries. QASYIPJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
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** Named semaphore close or shared memory detach
157	225	611	IPC Type	Char(1)	IPC Type **M** Shared memory **N** Named 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

Table 182. IP (Interprocess Communication) journal entries. QASYIPJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
205	273	659	Group Authority	Char(3)	Group's authority to IPC entity ***R** read ***W** write ***RW** read and write
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 two characters contain the length of the semaphore name.

IR (IP Rules Actions) journal entries

This table provides the format of the IR (IP Rules Actions) journal entries.

Table 183. IR (IP Rules Actions) journal entries. QASYIRJ4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630 and"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632 for field listing.

Table 183. IR (IP Rules Actions) journal entries. QASYIRJ4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	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)	
	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 path name.

Table 183. IR (IP Rules Actions) journal entries. QASYIRJ4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
	896	1282	Path Name Country or Region ID	Char(2)	The Country or Region ID for the path name.
	898	1284	Path Name Language ID	Char(3)	The language ID for the path name.
	901	1287	Path Name Length	Binary(4)	The length of the path name.
	903	1289	Path Name Indicator	Char(1)	Path name indicator: **Y** The Path Name field contains complete absolute path name for the object. **N** The Path Name field does not contain an absolute path name for the object, instead it contains a relative path name. The Relative Directory File ID field is valid and can be used to form an absolute path name with this relative path name.
	904	1290	Relative Directory File ID[3]	Char(16)	When the Path Name Indicator field is N, this field contains the file ID of the directory that contains the object identified in the Path Name field. Otherwise it contains hex zeros.[3]
	920	1306	Path Name[4]	Char(5002)	The path name of the object.

1
These fields are used only for objects in the "root" (/), QOpenSys, and user-defined 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
If the Path Name Indicator field is N, but the Relative Directory File ID is hex zeros, then there was some error in determining the path name information.

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.

IS (Internet Security Management) journal entries

This table provides the format of the IS (Internet Security Management) journal entries.

Table 184. IS (Internet Security Management) journal entries. QASYISJ4/J5 Field Description File

Offset			Field	Format	Description
JE	**J4**	**J5**			
	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630 and "Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632 for field listing.
	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[1]	Char(15)	Local IP Address.
	240	626	Local Client ID Port	Char(5)	Local Client ID port.
	245	631	Remote IP Address[1]	Char (15)	Remote IP address.
	260	646	Remote Client ID Port	Char (5)	Remote Client ID Port (valid for phase 2).
	265	651	Local IP Address Family	Char (1)	Local IP address family **4** IPv4 **6** IPv6
		652	Local IP Address	Char (46)	Local IP address
		698	Remote IP Address Family	Char (1)	Remote IP address family **4** IPv4 **6** IPv6
		699	Remote IP Address	Char (46)	Remote IP address

Table 184. IS (Internet Security Management) journal entries. QASYISJ4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		745	IKE Version	Char(4)	IKE version
		749	Reserved	Char(158)	Reserved
	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** IBM i 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 **5** IP version 6 address **6** IP version 6 subnet **7** IP version 4 address range **8** IP version 6 address range **9** Distinguished name **11** Key identifier
	787	1173	Local Client ID Value	Char(256)	Local client ID (valid for phase 2)

Table 184. IS (Internet Security Management) journal entries. QASYISJ4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
	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
	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 **5** IP version 6 address **6** IP version 6 subnet **7** IP version 4 address range **8** IP version 6 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)

1 This field only supports IPv4 addresses.

JD (Job Description Change) journal entries

This table provides the format of the JD (Job Description Change) journal entries.

Table 185. JD (Job Description Change) journal entries. QASYJDJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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 where the object is stored.
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

JS (Job Change) journal entries

This table provides the format of the JS (Job Change) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_JS table function: AUDIT_JOURNAL_JS

Table 186. JS (Job Change) journal entries. QASYJSJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See “Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)” on page 630, “Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)” on page 632, and “Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)” on page 633 for field listing.

Table 186. JS (Job Change) journal entries. QASYJSJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
156	224	610	Entry Type	Char(1)	The type of entry. **A** ENDJOBABN command **B** Submit **C** Change **E** End **H** Hold **I** Disconnect **J** The current job is attempting to interrupt another job **K** The current job is about to be interrupted **L** The interruption of the current job has completed **M** Change profile or group profile **N** ENDJOB command **P** Attach prestart or batch immediate job **Q** Change query attributes **R** Release **S** Start **T** Change profile or group profile using a profile token. **U** CHGUSRTRC **V** Virtual device changed by QWSACCDS API.

Table 186. JS (Job Change) journal entries. QASYJSJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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
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 device 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

Table 186. JS (Job Change) journal entries. QASYJSJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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.
	964	1350	Real Group Profile	Char(10)	The name of the real group profile for the thread.
	974	1360	Saved Group Profile	Char(10)	The name of the saved group profile for the thread.

Table 186. JS (Job Change) journal entries. QASYJSJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	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

Table 186. JS (Job Change) journal entries. QASYJSJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		3641	ASP name	Char(10)	ASP name for JOBD library
		3651	ASP number	Char(5)	ASP number for JOBD library
		3656	Time Zone Name	Char(10)	The time zone description name
		3666	Exit Job Name or Workload Capping Group Name[6, 7, 8]	Char(10)	Can contain any of the following values: • The name of the job that interrupted the current job • The name of the job that was interrupted by the current job • The name of the workload capping group associated with the job
		3676	Exit Job User	Char(10)	The user of the job that interrupted the current job, or the user of the job that was interrupted by the current job
		3686	Exit Job Number[6, 7]	Char(6)	The number of the job that interrupted the current job, or the job number of the job that was interrupted by the current job
		3692	Exit Program Name[6]	Char(10)	The exit program used to interrupt the job
		3702	Exit Program Library[6]	Char(10)	The library name of the exit program used to interrupt the job
		3712	JOBQ Library ASP Name	Char(10)	ASP name for JOBQ library
		3722	JOBQ Library ASP Number	Char(5)	ASP numer of JOBQ library

Table 186. JS (Job Change) journal entries. QASYJSJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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 610) 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 729. **5** This is a variable length field. The first two bytes contain the length of the data in the field. **6** This field is used only when entry type (offset 610) is J, K, or L. **7** When the entry type is J, this field contains information about the job that will be interrupted. When the entry type is K or L, this field contains information about the job that requested the interruption of the current job. **8** When the entry type is C, E, or S, this field contains the Workload Capping Group Name.					

KF (Key Ring File) journal entries

This table provides the format of the KF (Key Ring File) journal entries.

Table 187. KF (Key Ring File) journal entries. QASYKFJ4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	1	1			Heading fields common to all entry types. See “Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)” on page 630 and“Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)” on page 632 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

Table 187. KF (Key Ring File) journal entries. QASYKFJ4/J5 Field Description File *(continued)*					
Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	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.

Table 187. KF (Key Ring File) journal entries. QASYKFJ4/J5 Field Description File *(continued)*					
Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	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.
	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 destination file name length.
	830	1216	Object Name CCSID	Binary(5)	Source or destination file name CCSID.
	834	1220	Object Name Country or Region ID	Char(2)	Source or destination file name Country or Region ID.
	836	1222	Object Name Language ID	Char(3)	Source or destination file name language ID.
	839	1225	Reserved	Char(3)	
	842	1228	Parent File ID	Char(16)	Source or destination parent directory file ID.
	858	1244	Object File ID	Char(16)	Source or destination directory file ID.
	874	1260	Object Name	Char(512)	Source or destination 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 path name.
	2449	2835	Path Name Country or Region ID	Char(2)	The Country or Region ID for the path name.

Table 187. KF (Key Ring File) journal entries. QASYKFJ4/J5 Field Description File *(continued)*					
Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
	2451	2837	Path Name Language ID	Char(3)	The language ID for the path name.
	2454	2840	Path Name Length	Binary(4)	The length of the path name.
	2456	2842	Path Name Indicator	Char(1)	Path name indicator: **Y** The Path Name field contains complete absolute path name for the key ring file. **N** The Path Name field does not contain an absolute path name for the object, instead it contains a relative path name. The Relative Directory File ID field is valid and can be used to form an absolute path name with this relative path name.
	2457	2843	Relative Directory File ID[2]	Char(16)	When the Path Name Indicator field is N, this field contains the file ID of the directory that contains the object identified in the Path Name field. Otherwise it contains hex zeros.[2]
	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 destination file.
	7491	7877	ASP Name	Char(10)	Source or destination file ASP name
	7501	7887	ASP Number	Char(5)	Source or destination file ASP number
	7506	7892	Path Name CCSID	Binary(5)	The coded character set identifier for the path name.
	7510	7896	Path name Country or Region ID	Char(2)	The Country or Region ID for the path name.
	7512	7898	Path Name Language ID	Char(3)	The language ID for the path name.
	7515	7901	Path Name Length	Binary(4)	The length of the path name.

Table 187. KF (Key Ring File) journal entries. QASYKFJ4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
	7517	7903	Path Name Indicator	Char(1)	Path name indicator: **Y** The Path Name field contains complete absolute path name for the source or destination file. **N** The Path Name field does not contain an absolute path name for the object, instead it contains a relative path name. The Relative Directory File ID field is valid and can be used to form an absolute path name with this relative path name.
	7518	7904	Relative Directory File ID[3]	Char(16)	When the Path Name Indicator field is N, this field contains the file ID of the directory that contains the object identified in the Path Name field. Otherwise it contains hex zeros.[2]
	7534	7920	Absolute Path Name[1]	Char(5002)	The absolute path name of the source or destination file.

1
This is a variable length field. The first 2 bytes contain the length of the path name.

2
If the Path Name Indicator field is N, but the Relative Directory File ID is hex zeros, then there was some error in determining the path name information.

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 zeros.

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.

LD (Link, Unlink, Search Directory) journal entries

This table provides the format of the LD (Link, Unlink, Search Directory) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_LD table function: AUDIT_JOURNAL_LD

Table 188. LD (Link, Unlink, Search Directory) journal entries. QASYLDJE/J4/J5 Field Description File

Offset			Field	Format	Description
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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.
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 path name.

Table 188. LD (Link, Unlink, Search Directory) journal entries. QASYLDJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
	836	1222	Path Name Country or Region ID	Char(2)	The Country or Region ID for the path name.
	838	1224	Path Name Language ID	Char(3)	The language ID for the path name.
	841	1227	Path Name Length	Binary(4)	The length of the path name.
	843	1229	Path Name Indicator	Char(1)	Path name indicator: **Y** The Path Name field contains complete absolute path name for the object. **N** The Path Name field does not contain an absolute path name for the object, instead it contains a relative path name. The Relative Directory File ID field is valid and can be used to form an absolute path name with this relative path name.
	844	1230	Relative Direcotry File ID[1]	Char(16)	When the Path Name Indicator field is N, this field contains the file ID of the directory that contains the object identified in the Path Name field. Otherwise it contains hex zeros.[1]
	860	1246	Path Name[2]	Char(5002)	The path name of the object.

1
If the Path Name Indicator field is N, but the Relative Directory File ID is hex zeros, then there was some error in determining the path name information.

2
This is a variable length field. The first 2 bytes contain the length of the path name.

ML (Mail Actions) journal entries

This table provides the format of the ML (Mail Actions) journal entries.

Table 189. ML (Mail Actions) journal entries. QASYMLJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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.
167	235	621	User ID	Char(8)	User identifier
175	243	629	Address	Char(8)	User address

M0 (Db2 Mirror Setup Tools) journal entries

This table provides the format of the M0 (Db2 Mirror Setup Tools) journal entries. These journal entries are sent from the Db2 Mirror for i product.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_M0 table function: AUDIT_JOURNAL_M0

Table 190. M0 (Db2 Mirror Setup Tools) journal entries. QASYM0J5 Field Description File			
Offset			
J5	**Field**	**Format**	**Description**
1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630 for field listing.
610	Entry Type	Char(1)	The type of entry. **A** Db2 Mirror setup tools

Table 190. M0 (Db2 Mirror Setup Tools) journal entries. QASYM0J5 Field Description File *(continued)*

Offset J5	Field	Format	Description
611	Action	Char(15)	The action performed. **START** Begin the entire SYSBASE cloning process. **POWEROFF** Power off the setup source or copy node using the HMC poweroff operation. **PRECHECK** Perform validation and checking to ensure the whole cloning process will complete successfully. **FLASHCOPY** Perform the flash copy process on the storage. **REMOTECOPY** Perform the remote copy process on the storage. **IASPCOPY** Perform the entire automated DB IASP clone process. **PREIASP** Perform the pre-IASP copy steps. **POSTIASP** Perform the post-IASP copy steps. **STARTWARMCLONE** Start Db2 Mirror tracking and flush main memory on the setup source node. **CHECKSYSBASE** Verify the cloning of SYSBASE and the configuration of the setup copy node has completed successfully. **CONFIGFILE** Manipulate the JSON configuration files.

Table 190. M0 (Db2 Mirror Setup Tools) journal entries. QASYM0J5 Field Description File *(continued)*

Offset J5	Field	Format	Description
626	Action Type	Char(10)	The type of action being performed. When the Action (J5 offset 611) is START this field can contain: **WARM** **COLD** When the Action is POWEROFF this field can contain: **HMC** **CONTROL** **IMMED** When the Action is IASPCOPY this field can contain: **WARM** **COLD** When the Action is CONFIGFILE this field can contain: **UPDATE** **NEW** **SAVE** **RESTORE**
636	Status	Char(1)	Status of the action. This field may only contain data when the Action (J5 offset 611) is START, CHECKSYSBASE, POWEROFF, PRECHECK, FLASHCOPY, REMOTECOPY, and IASPCOPY. For these Actions, two audit entries will be sent. One when the Action starts and another when the action ends. When the audit entry is for the start of the Action this field will be blank. When the audit entry is for the end of the Action this field will contain the status of the action. **Y** The action was successful **N** The actions was not successful
637	ASP Name	Char(10)	ASP name. This field will contain data when Action (J5 offset 611) is IASPCOPY, PREIASP, or POSTIASP.
647	(Reserved area)	Char(15)	
662	Setup Source Node	Char(8)	The partition name of the Db2 Mirror setup source node.
670	Setup Copy Node	Char(8)	The partition name of the Db2 Mirror setup copy node.
678	Setup Source Storage	Char(256)	The IP address or host and domain name of the setup source storage system. This field will contain data when Action (J5 offset 611) is START, PRECHECK, FLASHCOPY, REMOTECOPY, or IASPCOPY.

Table 190. M0 (Db2 Mirror Setup Tools) journal entries. QASYM0J5 Field Description File *(continued)*			
Offset J5	**Field**	**Format**	**Description**
934	Setup Copy Storage	Char(256)	The IP address or host and domain name of the setup copy storage system. This field will contain data when Action (J5 offset 611) is START, PRECHECK, FLASHCOPY, REMOTECOPY, or IASPCOPY.

M6 (Db2 Mirror Communication Services) journal entries

This table provides the format of the M6 (Db2 Mirror Communication Services) journal entries. These journal entries are sent from the Db2 Mirror for i product.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_M6 table function: AUDIT_JOURNAL_M6

Table 191. M6 (Db2 Mirror Communication Services) journal entries. QASYM6J5 Field Description File			
Offset J5	**Field**	**Format**	**Description**
1			Heading fields common to all entry types. See “Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)” on page 630 for field listing.
610	Entry Type	Char(1)	The type of entry. **A** Add Network Redundancy Group (NRG) **C** Change NRG **R** Remove NRG
611	Name	Char(16)	The name associated with the NRG.
627	Type	Char(1)	Type of NRG. **'0'** Unspecified group **'1'** Db2 Mirror group **'2'** User defined group
628	Description	Char(50)	Text description.
678	(Reserved Area)	Char(16)	
694	Load Balance Link Count	Bin(5)	Load balance link count.
698	Pair Count	Bin(5)	Number of links configured in the group.

Table 191. M6 (Db2 Mirror Communication Services) journal entries. QASYM6J5 Field Description File *(continued)*

Offset J5	Field	Format	Description
702	Link 1 Address Family	Char(10)	The address family of the first link. ***IPV4** Internet Protocol Version 4 ***IPV6** Internet Protocol Version 6
712	Link 1 Local Address	Char(46)	The local IP address for the first link.
758	Link 1 Local Line Description	Char(16)	The local line description for the first link.
774	Link 1 Local VLAN ID	Bin(5)	The local VLAN ID for the first link.
778	Link 1 Remote address	Char(46)	The remote IP address for the first link.
824	Link 1 Type	Char(2)	The link type of the first link. **V1** RoCE v1 **V2** RoCE v2 **S2** Secure RoCE v2
826	Link 1 Pair Priority	Bin(5)	The priority of the first link.
830	Link 2 Address Family	Char(10)	The address family of the second link. This field can contain the same values as Link 1 Address Family (J5 offset 702).
840	Link 2 Local Address	Char(46)	The local IP address for the second link.
886	Link 2 Local Line Description	Char(16)	The local line description for the second link.
902	Link 2 Local VLAN ID	Bin(5)	The local VLAN ID for the second link.
906	Link 2 Remote address	Char(46)	The remote IP address for the second link.
952	Link 2 Type	Char(2)	The link type of the second link. This field can contain the same values as Link 1 Type (J5 offset 824).
954	Link 2 Pair Priority	Bin(5)	The priority of the second link.
958	Link 3 Address Family	Char(10)	The address family of the third link. This field can contain the same values as Link 1 Address Family (J5 offset 702).
968	Link 3 Local Address	Char(46)	The local IP address for the third link.

Table 191. M6 (Db2 Mirror Communication Services) journal entries. QASYM6J5 Field Description File *(continued)*

Offset			
J5	**Field**	**Format**	**Description**
1014	Link 3 Local Line Description	Char(16)	The local line description for the third link.
1030	Link 3 Local VLAN ID	Bin(5)	The local VLAN ID for the third link.
1034	Link 3 Remote address	Char(46)	The remote IP address for the third link.
1080	Link 3 Type	Char(2)	The link type of the third link. This field can contain the same values as Link 1 Type (J5 offset 824).
1082	Link 3 Pair Priority	Bin(5)	The priority of the third link.
1086	Link 4 Address Family	Char(10)	The address family of the fourth link. This field can contain the same values as Link 1 Address Family (J5 offset 702).
1096	Link 4 Local Address	Char(46)	The local IP address for the fourth link.
1142	Link 4 Local Line Description	Char(16)	The local line description for the fourth link.
1158	Link 4 Local VLAN ID	Bin(5)	The local VLAN ID for the fourth link.
1162	Link 4 Remote address	Char(46)	The remote IP address for the fourth link.
1208	Link 4 Type	Char(2)	The link type of the fourth link. This field can contain the same values as Link 1 Type (J5 offset 824).
1210	Link 4 Pair Priority	Bin(5)	The priority of the fourth link.
1214	Link 5 Address Family	Char(10)	The address family of the fifth link. This field can contain the same values as Link 1 Address Family (J5 offset 702).
1224	Link 5 Local Address	Char(46)	The local IP address for the fifth link.
1270	Link 5 Local Line Description	Char(16)	The local line description for the fifth link.
1286	Link 5 Local VLAN ID	Bin(5)	The local VLAN ID for the fifth link.
1290	Link 5 Remote address	Char(46)	The remote IP address for the fifth link.
1336	Link 5 Type	Char(2)	The link type of the fifth link. This field can contain the same values as Link 1 Type (J5 offset 824).
1338	Link 5 Pair Priority	Bin(5)	The priority of the fifth link.
1342	Link 6 Address Family	Char(10)	The address family of the sixth link. This field can contain the same values as Link 1 Address Family (J5 offset 702).

Table 191. M6 (Db2 Mirror Communication Services) journal entries. QASYM6J5 Field Description File *(continued)*

Offset J5	Field	Format	Description
1352	Link 6 Local Address	Char(46)	The local IP address for the sixth link.
1398	Link 6 Local Line Description	Char(16)	The local line description for the sixth link.
1414	Link 6 Local VLAN ID	Bin(5)	The local VLAN ID for the sixth link.
1418	Link 6 Remote address	Char(46)	The remote IP address for the sixth link.
1464	Link 6 Type	Char(2)	The link type of the sixth link. This field can contain the same values as Link 1 Type (J5 offset 824).
1466	Link 6 Pair Priority	Bin(5)	The priority of the sixth link.
1470	Link 7 Address Family	Char(10)	The address family of the seventh link. This field can contain the same values as Link 1 Address Family (J5 offset 702).
1480	Link 7 Local Address	Char(46)	The local IP address for the seventh link.
1526	Link 7 Local Line Description	Char(16)	The local line description for the seventh link.
1542	Link 7 Local VLAN ID	Bin(5)	The local VLAN ID for the seventh link.
1546	Link 7 Remote address	Char(46)	The remote IP address for the seventh link.
1592	Link 7 Type	Char(2)	The link type of the seventh link. This field can contain the same values as Link 1 Type (J5 offset 824).
1594	Link 7 Pair Priority	Bin(5)	The priority of the seventh link.
1598	Link 8 Address Family	Char(10)	The address family of the eighth link. This field can contain the same values as Link 1 Address Family (J5 offset 702).
1608	Link 8 Local Address	Char(46)	The local IP address for the eighth link.
1654	Link 8 Local Line Description	Char(16)	The local line description for the eighth link.
1670	Link 8 Local VLAN ID	Bin(5)	The local VLAN ID for the eighth link.
1674	Link 8 Remote address	Char(46)	The remote IP address for the eighth link.
1720	Link 8 Type	Char(2)	The link type of the eighth link. This field can contain the same values as Link 1 Type (J5 offset 824).
1722	Link 8 Pair Priority	Bin(5)	The priority of the eighth link.

Table 191. M6 (Db2 Mirror Communication Services) journal entries. QASYM6J5 Field Description File *(continued)*

Offset J5	Field	Format	Description
1726	Link 9 Address Family	Char(10)	The address family of the ninth link. This field can contain the same values as Link 1 Address Family (J5 offset 702).
1736	Link 9 Local Address	Char(46)	The local IP address for the ninth link.
1782	Link 9 Local Line Description	Char(16)	The local line description for the ninth link.
1798	Link 9 Local VLAN ID	Bin(5)	The local VLAN ID for the ninth link.
1802	Link 9 Remote address	Char(46)	The remote IP address for the ninth link.
1848	Link 9 Type	Char(2)	The link type of the ninth link. This field can contain the same values as Link 1 Type (J5 offset 824).
1850	Link 9 Pair Priority	Bin(5)	The priority of the ninth link.
1854	Link 10 Address Family	Char(10)	The address family of the tenth link. This field can contain the same values as Link 1 Address Family (J5 offset 702).
1864	Link 10 Local Address	Char(46)	The local IP address for the tenth link.
1910	Link 10 Local Line Description	Char(16)	The local line description for the tenth link.
1926	Link 10 Local VLAN ID	Bin(5)	The local VLAN ID for the tenth link.
1930	Link 10 Remote address	Char(46)	The remote IP address for the tenth link.
1976	Link 10 Type	Char(2)	The link type of the tenth link. This field can contain the same values as Link 1 Type (J5 offset 824).
1978	Link 10 Pair Priority	Bin(5)	The priority of the tenth link.
1982	Link 11 Address Family	Char(10)	The address family of the eleventh link. This field can contain the same values as Link 1 Address Family (J5 offset 702).
1992	Link 11 Local Address	Char(46)	The local IP address for the eleventh link.
2038	Link 11 Local Line Description	Char(16)	The local line description for the eleventh link.
2054	Link 11 Local VLAN ID	Bin(5)	The local VLAN ID for the eleventh link.
2058	Link 11 Remote address	Char(46)	The remote IP address for the eleventh link.

Table 191. M6 (Db2 Mirror Communication Services) journal entries. QASYM6J5 Field Description File *(continued)*

Offset J5	Field	Format	Description
2104	Link 11 Type	Char(2)	The link type of the eleventh link. This field can contain the same values as Link 1 Type (J5 offset 824).
2106	Link 11 Pair Priority	Bin(5)	The priority of the eleventh link.
2110	Link 12 Address Family	Char(10)	The address family of the twelfth link. This field can contain the same values as Link 1 Address Family (J5 offset 702).
2120	Link 12 Local Address	Char(46)	The local IP address for the twelfth link.
2166	Link 12 Local Line Description	Char(16)	The local line description for the twelfth link.
2182	Link 12 Local VLAN ID	Bin(5)	The local VLAN ID for the twelfth link.
2186	Link 12 Remote address	Char(46)	The remote IP address for the twelfth link.
2232	Link 12 Type	Char(2)	The link type of the twelfth link. This field can contain the same values as Link 1 Type (J5 offset 824).
2234	Link 12 Pair Priority	Bin(5)	The priority of the twelfth link.
2238	Link 13 Address Family	Char(10)	The address family of the thirteenth link. This field can contain the same values as Link 1 Address Family (J5 offset 702).
2248	Link 13 Local Address	Char(46)	The local IP address for the thirteenth link.
2294	Link 13 Local Line Description	Char(16)	The local line description for the thirteenth link.
2310	Link 13 Local VLAN ID	Bin(5)	The local VLAN ID for the thirteenth link.
2314	Link 13 Remote Address	Char(46)	The remote IP address for the thirteenth link.
2360	Link 13 Type	Char(2)	The link type of the thirteenth link. This field can contain the same values as Link 1 Type (J5 offset 824).
2362	Link 13 Pair Priority	Bin(5)	The priority of the thirteenth link.
2366	Link 14 Address Family	Char(10)	The address family of the fourteenth link. This field can contain the same values as Link 1 Address Family (J5 offset 702).
2376	Link 14 Local Address	Char(46)	The local IP address for the fourteenth link.
2422	Link 14 Local Line Description	Char(16)	The local line description for the fourteenth link.

Table 191. M6 (Db2 Mirror Communication Services) journal entries. QASYM6J5 Field Description File *(continued)*

Offset J5	Field	Format	Description
2438	Link 14 Local VLAN ID	Bin(5)	The local VLAN ID for the fourteenth link.
2442	Link 14 Remote Address	Char(46)	The remote IP address for the fourteenth link.
2488	Link 14 Type	Char(2)	The link type of the fourteenth link. This field can contain the same values as Link 1 Type (J5 offset 824).
2490	Link 14 Pair Priority	Bin(5)	The priority of the fourteenth link.
2494	Link 15 Address Family	Char(10)	The address family of the fifteenth link. This field can contain the same values as Link 1 Address Family (J5 offset 702).
2504	Link 15 Local Address	Char(46)	The local IP address for the fifteenth link.
2550	Link 15 Local Line Description	Char(16)	The local line description for the fifteenth link.
2566	Link 15 Local VLAN ID	Bin(5)	The local VLAN ID for the fifteenth link.
2570	Link 15 Remote Address	Char(46)	The remote IP address for the fifteenth link.
2616	Link 15 Type	Char(2)	The link type of the fifteenth link. This field can contain the same values as Link 1 Type (J5 offset 824).
2618	Link 15 Pair Priority	Bin(5)	The priority of the fifteenth link.
2622	Link 16 Address Family	Char(10)	The address family of the sixteenth link. This field can contain the same values as Link 1 Address Family (J5 offset 702).
2632	Link 16 Local Address	Char(46)	The local IP address for the sixteenth link.
2678	Link 16 Local Line Description	Char(16)	The local line description for the sixteenth link.
2694	Link 16 Local VLAN ID	Bin(5)	The local VLAN ID for the sixteenth link.
2698	Link 16 Remote Address	Char(46)	The remote IP address for the sixteenth link.
2744	Link 16 Type	Char(2)	The link type of the sixteenth link. This field can contain the same values as Link 1 Type (J5 offset 824).
2746	Link 16 Pair Priority	Bin(5)	The priority of the sixteenth link.

M7 (Db2 Mirror Replication Services) journal entries

This table provides the format of the M7 (Db2 Mirror Replication Services) journal entries. These journal entries are sent from the Db2 Mirror for i product.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_M7 table function: AUDIT_JOURNAL_M7

Table 192. M7 (Db2 Mirror Replication Services) journal entries. QASYM7J5 Field Description File

Offset J5	Field	Format	Description
1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630 for field listing.
610	Entry Type	Char(1)	The type of entry. **A** Add active replication criteria rule **D** Duplicate replication criteria rules (a rename library was performed) **P** Activate pending replication criteria rules **R** Remove active replication criteria rule **S** Resynchronization of eligible objects **U** User deferred or deleted entries in the Object Tracking List (OTL) using the SQL QSYS2.CHANGE_RESYNC_ENTRIES procedure **V** Generic versioning
611	Action	Char(3)	Action to perform. When Entry type (J5 offset 610) is V this field can contain: **ADD** Register (add) an applied version information entry for a specific feature or function in the Mirror Version List (MVL). **APY** Apply pending version information entries. **RMV** Unregister (remove) an applied version information entry from the Mirror Version List (MVL). **RFS** Refresh the version information entries in Mirror Version List by running user specified Version Handlers.

Table 192. M7 (Db2 Mirror Replication Services) journal entries. QASYM7J5 Field Description File *(continued)*

Offset J5	Field	Format	Description
614	Rule Identifier	Bin(5)	When Entry type (J5 offset 610) is A, D, or R this is the identifier for this replication criteria rule. When Entry type (J5 offset 610) is V and Action (J5 offset 611) is ADD or RMV, this is the version entry number.
618	Resync Type	Char(10)	When Entry type (J5 offset 610) is S this is the type of resynchronization performed. **RESUME** Resynchronization of objects that are on the Object Tracking List (OTL) because the node was previously blocked. **RECLONE** Resynchronization of actively replicating objects. When Entry type (J5 offset 610) is V this is the version group.
628	Inclusion State	Char(10)	When Entry type (J5 offset 610) is A or D this is the inclusion state of the replication criteria rule. **DEFINITION** Objects that best match this replication criteria rule are replicated. Only the definition of the object is replicated. **EXCLUDE** Objects that best match this replication criteria rule are not replicated **INCLUDE** Objects that best match this replication criteria rule are replicated When Entry type (J5 offset 610) is V and Action (J5 offset 611) is ADD or RMV this is the version entry activation state indicator. **IMMEDIATE** This version entry can be activated immediately from an applied state. **RESUME** This version entry can be activated from an applied state the next time replication is resumed.
638	ASP Name	Char(10)	ASP Name or *SYSBAS. When entry type (J5 offset 610) is A, D, P, S, or R this field contains the name of the ASP associated with this replication criteria rule. This field will be blank if System value (J5 offset 686) or Environment variable (J5 offset 696) contain a value. When entry type (J5 offset 610) is U this field contains the ASP associated with the SQL QSYS2.CHANGE_RESYNC_ENTRIES procedure.

Table 192. M7 (Db2 Mirror Replication Services) journal entries. QASYM7J5 Field Description File *(continued)*

Offset J5	Field	Format	Description
648	Library Name	Char(10)	The library name. When entry type (J5 offset 610) is A, D, or R this field contains the name of the library associated with the replication criteria rule. If this field is blank and ASP name (J5 offset 638) contains a value, then all supported objects in all libraries in the ASP will be operated on. When entry type (J5 offset 610) is U this field contains the name of the library associated with the SQL QSYS2.CHANGE_RESYNC_ENTRIES procedure.
658	Object Type	Char(8)	The object type. When Entry type (J5 offset 610) is A, D, or R this is the object type associated with the replication criteria rule. If this field is blank and Library name (J5 offset 648) contains a value, then all objects of all supported object types in that library will be operated on. When Entry type (J5 offset 610) is U this is the object type associated with the SQL QSYS2.CHANGE_RESYNC_ENTRIES procedure.
666	Object Name	Char(10)	The name of the object. When entry type (J5 offset 610) is A, D, or R this field contains the name of the object associated with this replication criteria rule. If this field is blank and Library name (J5 offset 648) contains a value, then all objects of Object type (J5 offset 658) in that library will be operated on. When entry type (J5 offset 610) is U this field contains the name of the object associated with the SQL QSYS2.CHANGE_RESYNC_ENTRIES procedure.
676	Original Library Name	Char(10)	Original library name. This field only contains data when Entry type is D.
686	System Value	Char(10)	The name of the system value. When Entry type (J5 offset 610) is A or R this field contains the system value associated with this replication criteria rule. This field will be blank if Object name (J5 offset 666) or Environment variable (J5 offset 696) contain a value.
696	Environment Variable	Char(128)	When Entry type (J5 offset 610) is A or R this field contains the environment variable associated with this replication criteria rule. This field will be blank if Object name (J5 offset 666) or System value (J5 offset 686) contains a value. When Entry type (J5 offset 610) is V this field contains the version name. If this field is blank and Version Group (J5 offset 618) contains a value, all version names in that group will be operated on.

Table 192. M7 (Db2 Mirror Replication Services) journal entries. QASYM7J5 Field Description File *(continued)*			
Offset J5	**Field**	**Format**	**Description**
824	Apply Label	Char(26)	When Entry type (J5 offset 610) is A, P, or R this is the label used to identify replication criteria rules. When Entry type (J5 offset 610) is V and Action (J5 offset 611) is ADD or RMV this is the version identifier. The format is xxx.yyy.zzz where each piece of the version contains the digits 0-9: **xxx** The major version number. This value is always present for a version number. **yyy** An optional minor version number. **zzz** An optional revision number.
850	(Reserved Area)	Char(2)	
852	Number of Objects	Bin(5)	When Entry type (J5 offset 610) is A, P, R, or S this is the number of Save/Restore entries added to the OTL for objects affected by this operation. When Entry type (J5 offset 610) is U this is the number of OTL rows affected by the SQL QSYS2.CHANGE_RESYNC_ENTRIES procedure.

M8 (Db2 Mirror Product Services) journal entries

This table provides the format of the M8 (Db2 Mirror Product Services) journal entries. These journal entries are sent from the Db2 Mirror for i product.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_M8 table function: AUDIT_JOURNAL_M8

Table 193. M8 (Db2 Mirror Product Services) journal entries. QASYM8J5 Field Description File			
Offset J5	**Field**	**Format**	**Description**
1			Heading fields common to all entry types. See “Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)” on page 630 for field listing.

Table 193. M8 (Db2 Mirror Product Services) journal entries. QASYM8J5 Field Description File *(continued)*			
Offset J5	**Field**	**Format**	**Description**
610	Entry Type	Char(1)	The type of entry. **A** Add IASP **C** Change mirror **F** Change flight recorder **I** Set default inclusion state **J** Change mirror ObjectConnect **L** Reclone replicated objects **O** Takeover **R** Remove IASP **S** Setup mirror **T** Terminate mirror **W** Swap mirror roles
611	ASP Name	Char(10)	When Entry type (J5 offset 610) is A (Add IASP), C (Change mirror), O (Takeover and Action (J5 offset 649) is CREATE or CHANGE), or R (Remove IASP) this field contains the ASP name or *SYSBAS.
621	IASP Type	Char(8)	When Entry type (J5 offset 610) is A (Add IASP) or R (Remove IASP) this field contains the type of IASP. **IFS** This IASP is for an IFS ASP group. **DATABASE** This IASP is for a database ISAP group.

Table 193. M8 (Db2 Mirror Product Services) journal entries. QASYM8J5 Field Description File *(continued)*

Offset J5	Field	Format	Description
629	Default Inclusion State	Char(10)	When the Entry type (J5 offset 610) is A (Add IASP) this field contains the default object inclusion state for objects in this ASP. When Entry type (J5 offset 610) is I (Set default inclusion state) this field contains the default inclusion state for objects in *SYSBAS. **EXCLUDE** Eligible objects not covered by an RCL rule will not be replicated. **INCLUDE** Eligible objects not covered by an RCL rule will be replicated. **RESET** Clear the default inclusion state. This value applies when Entry type (J5 offset 610) is I.
639	Cluster Resource Group	Char(10)	When Entry type (J5 offset 610) is A (Add IASP) this field contains the cluster resource group name. When Entry type (J5 offset 610) is O (Takeover) this field contains the takeover group name.

Table 193. M8 (Db2 Mirror Product Services) journal entries. QASYM8J5 Field Description File *(continued)*

Offset J5	Field	Format	Description
649	Action	Char(10)	Action to perform. When Entry type (J5 offset 610) is A (Add IASP) this field can contain: **NEW** The IASP is being defined for the first time. **RECLONE** The IASP is used as the source of a reclone operation. **SHADOW** The IASP is being pre-defined as an IASP on a PowerHA® disaster recovery system. When Entry type (J5 offset 610) is C (Change mirror) this field can contain: **DISABLE** Disable. **ENABLE** Enable. **RESUME** Resume replication. **RESUMEABN** Abnormal resume replication. **SUSPEND** Suspend replication. **SUSPMAINT** Suspend for maintenance. **REQUIRED** Required. **NTREQUIRED** Not required. When Entry type (J5 offset 610) is F (Change flight recorder) this field can contain: **ENDJOB** End flight recorder QMRDBLOGR job. **STARTJOB** Start flight recorder QMRDBLOGR job. **SUSPEND** Suspend flight recorder logging. **RESUME** Resume flight recorder logging.

Table 193. M8 (Db2 Mirror Product Services) journal entries. QASYM8J5 Field Description File *(continued)*

Offset J5	Field	Format	Description
			When Entry type (J5 offset 610) is J (Change mirror ObjectConnect) this field can contain: **END** End ObjectConnect Server. **START** Start ObjectConnect Server. **CHANGE** Change ObjectConnect Server. When Entry type (J5 offset 610) is L (Reclone replicated objects) this field can contain: **RESUMEABN** Reclone replicated objects with abnormal resume. When Entry type (J5 offset 610) is O (Takeover) this field can contain: **DELETE** Delete mirror takeover group. **CREATE** Create mirror takeover group. **CHANGE** Change mirror takeover group. **SWAP** Swap mirror takeover group. **ADD** Add mirror takeover address. **REMOVE** Remove mirror takeover address. When Entry type (J5 offset 610) is T (Terminate mirror) this field can contain: **RECLONE** Active replication is ended. **DESTROY** Db2 Mirror is ended.

Table 193. M8 (Db2 Mirror Product Services) journal entries. QASYM8J5 Field Description File *(continued)*

Offset J5	Field	Format	Description
659	Auto Resume	Char(1)	When Entry type (J5 offset 610) is C (Change mirror) this field specifies whether to automatically resume mirroring. This field can contain: **Y** Automatically resume mirroring after being suspended. **N** Do not automatically resume mirroring. When Entry type (J5 offset 610) is J (Change mirror ObjectConnect) this field specifies whether to automatically start the ObjectConnect for Db2 Mirror server. This field can contain: **Y** Automatically start the ObjectConnect for Db2 Mirror server. **N** Do not automatically start the ObjectConnect for Db2 Mirror server. When Entry type (J5 offset 610) is O (Takeover) and Action (J5 offset 649) is CREATE or CHANGE this field specifies whether the takeover IP address group should be automatically switched back to its preferred node. This field can contain: **Y** Automatically return this takeover IP address group to its preferred node. **N** Do not automatically return this takeover IP address group.
660	Auto Swap	Char(1)	When Entry type (J5 offset 610) is C (Change mirror) this field contains the swap behavior on power down system. **Y** Automatically swap roles. **N** Do not automatically swap roles.
661	Parallel Degree	Char(5)	When Entry type (J5 offset 610) is C (Change mirror) this field contains the degree of parallelism to be used for Db2 Mirror resynchronization processing. This field may contain NONE. When Entry type (J5 offset 610) is J (Change mirror ObjectConnect) this field contains a character representation of the inactive time. The length of time, in minutes, that a server job will stay inactive before ending.

Table 193. M8 (Db2 Mirror Product Services) journal entries. QASYM8J5 Field Description File *(continued)*

Offset J5	Field	Format	Description
666	Primary Node	Char(8)	When Entry type (J5 offset 610) is S (Setup mirror) or T (Terminate mirror) this field contains the name of the partition designated as the primary node. When Entry type (J5 offset 610) is W (Swap mirror roles) this field contains the name of the new primary node. When Entry type (J5 offset 610) is O (Takeover) and Action (J5 offset 649) is CREATE or CHANGE this field contains the name of the preferred node.
674	Secondary Node	Char(8)	When Entry type (J5 offset 610) is S (Setup mirror), W (Swap mirror roles), or T (Terminate mirror) this field contains the name of the partition designated as the secondary node. When Entry type (J5 offset 610) is W this is the name of the new secondary node.
682	IP Address	Char(48)	When Entry type (J5 offset 610) is O (Takeover) and Action (J5 offset 649) is ADD or REMOVE this field contains the takeover IP address.
730	Archive Retention	Char(3)	When Entry type (J5 offset 610) is F (Change flight recorder) this field contains the number of days the flight recorder logs are retained. When Entry type (J5 offset 610) is J (Change mirror ObjectConnect) this field contains the minimum number of server jobs that are started.
733	Percent *SYSBAS for logs	Char(6)	When Entry type (J5 offset 610) is F (Change flight recorder) this field contains the percentage of *SYSBAS allocated for flight recorder logs. When Entry type (J5 offset 610) is C (Change mirror) this field contains the time, in seconds, that Db2 Mirror resynchronization processing should wait before looking for spooled files that need to be replicated. When Entry type (J5 offset 610) is J (Change mirror ObjectConnect) this field contains the maximum number of server jobs that are started.

Table 193. M8 (Db2 Mirror Product Services) journal entries. QASYM8J5 Field Description File *(continued)*

Offset J5	Field	Format	Description
739	Logging Category	Char(10)	When Entry type (J5 offset 610) is F (Change flight recorder) this field contains the category for which flight recorder entries will be logged. **ALL** All categories **CONFIG** Configuration processing **DATABASE** Database processing **DATAQ** Data queue handler **DBCONN** Database connection **ENGCOMM** Engine communication **ENGCONN** Engine connection **ENGCTRL** Engine controller **ENGJOB** Engine job **ENGSTATE** Engine state **FLIGHTREC** Flight recorder **HEALTHMON** Health monitor **IFSCONN** IFS connection **LOGGERTEST** Logger testing **NRG** Network redundancy groups **OBJCONN** Object connection **OBJRCVR** Object receiver **OBJREG** Object registry **OBJREPLMGR** Object replication manager **OBJSYNC** Object synchronization **RESYNC** Resynchronization

Table 193. M8 (Db2 Mirror Product Services) journal entries. QASYM8J5 Field Description File *(continued)*			
Offset J5	**Field**	**Format**	**Description**
			QUORUM quorum server **RCL** Replication criteria list **SAVRST** Save and restore processing **SECURITY** Security object handler **SPOOL** Spooled file handler **OUTQJOBQ** Output queue and job queue processing **UTILITIES** Utilities processing **VARYIASP** Vary IASP processing **WRKMGT** Work management When Entry type (J5 offset 610) is C (Change mirror) this field contains the product controls for Db2 Mirror. **ENCRPTRDMA** Encrypted RDMA **USERINDEX** *USRIDX object replication **USERSPACE** *USRSPC object replication **DTAQENTS** *DTAQ entries replication

Table 193. M8 (Db2 Mirror Product Services) journal entries. QASYM8J5 Field Description File *(continued)*			
Offset J5	**Field**	**Format**	**Description**
749	Logging Level	Char(5)	When Entry type (J5 offset 610) is F (Change flight recorder) this field contains the level at which an entry is written to the flight recorder log. **NONE** No log entries are generated. **ERROR** Log entries are generated for run time errors and unexpected conditions. **WARN** Log entries for the ERROR level are generated, plus entries for errors or other run time situations that are unexpected or unusual but not necessarily wrong. **INFO** Log entries for the WARN level are generated, plus interesting run time events. **DEBUG** Log entries for the INFO level are generated, plus debug information. **SYS** When Logging Category (J5 offset 739) is ALL the shipped default level is set for each category.

M9 (Db2 Mirror Replication State) journal entries

This table provides the format of the M9 (Db2 Mirror Replication State) journal entries. These journal entries are sent from the Db2 Mirror for i product.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_M9 table function: AUDIT_JOURNAL_M9

Table 194. M9 (Db2 Mirror Replication State) journal entries. QASYM9J5 Field Description File			
Offset J5	**Field**	**Format**	**Description**
1			Heading fields common to all entry types. See “Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)” on page 630 for field listing.
610	Entry Type	Char(1)	The type of entry. **C** Change to the replication state of an ASP
611	ASP Name	Char(10)	ASP name for which the replication state changed. This field may contain *SYSBAS.

Table 194. M9 (Db2 Mirror Replication State) journal entries. QASYM9J5 Field Description File *(continued)*			
Offset			
J5	**Field**	**Format**	**Description**
621	Replication State	Char(12)	Db2 Mirror replication state. **ACTIVE** **BLOCKED** **NOT MIRRORED** **TRACKING**
633	Previous Replication State	Char(12)	Previous Db2 Mirror replication state. **ACTIVE** **BLOCKED** **NOT MIRRORED** **TRACKING**
645	(Reserved Area)	Char(1)	
646	Reason for Change	Bin(5)	Reason the replication state changed to BLOCKED or TRACKING. For a description of the reason codes see Replication detail info. This field will only contain data when the Replication State (J5 offset 621) is BLOCKED or TRACKING.

NA (Attribute Change) journal entries

This table provides the format of the NA (Attribute Change) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_NA table function: AUDIT_JOURNAL_NA

Table 195. 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 "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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.

Table 195. NA (Attribute Change) journal entries. QASYNAJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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.

ND (APPN Directory Search Filter) journal entries

This table provides the format of the ND (APPN Directory Search Filter) journal entries.

Table 196. ND (APPN Directory Search Filter) journal entries. QASYNDJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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.

Table 196. ND (APPN Directory Search Filter) journal entries. QASYNDJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
205	273	659	Inbound session	Char(1)	Inbound session. **Y** This is an inbound session **N** This is not an inbound session
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 Protection of your system in an APPN and HPR environment for details.

NE (APPN End Point Filter) journal entries

This table provides the format of the NE (APPN End Point Filter) journal entries.

Table 197. NE (APPN End Point Filter) journal entries. QASYNEJE/J4/J5 Field Description File

Offset			Field	Format	Description
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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.

Table 197. NE (APPN End Point Filter) journal entries. QASYNEJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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 Protection of your system in an APPN and HPR environment for details.

OM (Object Management Change) journal entries

This table provides the format of the OM (Object Management Change) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_OM table function: AUDIT_JOURNAL_OM

Table 198. OM (Object Management Change) journal entries. QASYOMJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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.
167	235	621	Old Library Name	Char(10)	The name of the library in which the old object resides.
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.

Table 198. OM (Object Management Change) journal entries. QASYOMJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
195	263	649	New Library Name	Char(10)	The name of the library to which the object was moved.
205	273		(Reserved Area)	Char(20)	
		659	Object Attribute	Char(10)	The attribute of the object.
		669	(Reserved Area)	Char(10)	
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)	

Table 198. OM (Object Management Change) journal entries. QASYOMJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
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.
	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 path name.
	1618	2004	Path Name Country or Region ID	Char(2)	The Country or Region ID for the path name.
	1620	2006	Path Name Language ID	Char(3)	The language ID for the path name.
	1623	2009	Path Name Length	Binary(4)	The length of the path name.
	1625	2011	Path Name Indicator	Char(1)	Path name indicator: **Y** The Path Name field contains complete absolute path name for the object. **N** The Path Name field does not contain an absolute path name for the object, instead it contains a relative path name. The Relative Directory File ID field is valid and can be used to form an absolute path name with this relative path name.
	1626	2012	Relative Directory File ID[3]	Char(16)	When the Path Name Indicator field is N, this field contains the file ID of the directory that contains the object identified in the Path Name field. Otherwise it contains hex zeros.[3]
	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.

Table 198. OM (Object Management Change) journal entries. QASYOMJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
	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 path name.
	6679	7065	Path Name Country or Region ID	Char(2)	The Country or Region ID for the path name.
	6681	7067	Path Name Language ID	Char(3)	The language ID for the path name.
	6684	7070	Path Name Length	Binary(4)	The length of the path name.
	6686	7072	Path Name Indicator	Char(1)	Path name indicator: **Y** The Path Name field contains complete absolute path name for the object. **N** The Path Name field does not contain an absolute path name for the object, instead it contains a relative path name. The Relative Directory File ID field is valid and can be used to form an absolute path name with this relative path name.
	6687	7073	Relative Directory File ID[4]	Char(16)	When the Path Name Indicator field is N, this field contains the file ID of the directory that contains the object identified in the Path Name field. Otherwise it contains hex zeros.[3]
	6703	7089	Absolute Path Name[5]	Char(5002)	The new absolute path name of the object.

1
These fields are used only for objects in the "root" (/), QOpenSys, 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
If the Path Name Indicator field is N, but the Relative Directory File ID is hex zeros, then there was some error in determining the path name information.

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 zeros.

5
This is a variable length field. The first 2 bytes contain the length of the path name.

Table 198. OM (Object Management Change) journal entries. QASYOMJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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.					

OR (Object Restore) journal entries

This table provides the format of the OR (Object Restore) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_OR table function: AUDIT_JOURNAL_OR

Table 199. OR (Object Restore) journal entries. QASYORJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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.

Table 199. OR (Object Restore) journal entries. QASYORJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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)	
	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 IBM i format **E** Signature exists but is not verified **F** Signature does not match object content **I** Signature ignored **N** Unsignable object **S** Signature is valid **T** Untrusted signature **U** Object unsigned

Table 199. OR (Object Restore) journal entries. QASYORJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	278	664	Scan attribute	Char(1)	If the file was an integrated file system object, the value of the scan attribute for that object where **Y** *YES **N** *NO **C** *CHGONLY See the CHGATR command for descriptions of these values.
	279		(Reserved Area)	Char(14)	
		665	Object Attribute	Char(10)	The attribute of the object.
		675	(Reserved Area)	Char(4)	
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		(Reserved Area)	Char(18)	

Table 199. OR (Object Restore) journal entries. QASYORJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		865	Restore Private Authorities	Char(1)	Private authorities requested to be restored (PVTAUT(*YES) specified on restore command) **Y** PVTAUT(*YES) specified on restore command **N** PVTAUT(*NO) specified on restore command
		866	Private Authorities Saved[8]	Binary(5)	Number of private authorities saved
		870	Private Authorities Restored[8]	Binary(5)	Number of private authorities restored
		874	(Reserved Area)	Char(9)	
	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 identifer 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.
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.

Table 199. OR (Object Restore) journal entries. QASYORJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
	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 path name.
	1122	1508	Path Name Country or Region ID	Char(2)	The Country or Region ID for the path name.
	1124	1510	Path Name Language ID	Char(3)	The language ID for the path name.
	1127	1513	Path Name Length	Binary(4)	The length of the path name.
	1129	1515	Path Name Indicator	Char(1)	Path name indicator: **Y** The Path Name field contains complete absolute path name for the object. **N** The Path Name field does not contain an absolute path name for the object, instead it contains a relative path name. The Relative Directory File ID field is valid and can be used to form an absolute path name with this relative path name.
	1130	1516	Relative Directory File ID[5]	Char(16)	When the Path Name Indicator field is N, this field contains the file ID of the directory that contains the object identified in the Path Name field. Otherwise it contains hex zeros.[5]
	1146	1532	Path Name[6]	Char(5002)	The 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 This field is used only for objects in the "root" (/) ,QOpenSys, and user-defined 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 If the Path Name Indicator field is N, but the Relative Directory File ID is hex zeros, then there was some error in determining the path name information.

Table 199. *OR (Object Restore) journal entries.* QASYORJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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. **8** This field is zero if Restore Private Authorities (offset 865) is N.					

OW (Ownership Change) journal entries

This table provides the format of the OW (Ownership Change) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_OW table function: AUDIT_JOURNAL_OW

Table 200. *OW (Ownership Change) journal entries.* QASYOWJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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 where the object is stored.
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.

Table 200. OW (Ownership Change) journal entries. QASYOWJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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 path name.
	1007	1393	Path Name Country or Region ID	Char(2)	The Country or Region ID for the path name.
	1009	1395	Path Name Language ID	Char(3)	The language ID for the path name.
	1012	1398	Path Name Length	Binary(4)	The length of the path name.

Table 200. OW (Ownership Change) journal entries. QASYOWJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	1014	1400	Path Name Indicator	Char(1)	Path name indicator: **Y** The Path Name field contains complete absolute path name for the object. **N** The Path Name field does not contain an absolute path name for the object, instead it contains a relative path name. The Relative Directory File ID field is valid and may be used to form an absolute path name with this relative path name.
	1015	1401	Relative Directory File ID[3]	Char(16)	When the Path Name Indicator field is N, this field contains the file ID of the directory that contains the object identified in the Path Name field. Otherwise it contains hex zeros.[3]
	1031	1417	Path Name[4]	Char(5002)	The path name of the object.

1
These fields are used only for objects in the "root" (/), QOpenSys, and user-defined 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
If the Path Name Indicator field is N, but the Relative Directory File ID is hex zeros, then there was some error in determining the path name information.

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.

O1 (Optical Access) journal entries

This table provides the format of the O1 (Optical Access) journal entries.

Table 201. O1 (Optical Access) journal entries. QASY01JE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 for field listing.

Table 201. O1 (Optical Access) journal entries. QASY01JE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
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 S-Storage
158	226	612	Access Type	Char(1)	D-File Data A-File Directory Attributes R-Restore operation S-Save operation
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

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

O2 (Optical Access) journal entries

This table provides the format of the O2 (Optical Access) journal entries.

Table 202. O2 (Optical Access) journal entries. QASY02JE/J4/J5 Field Description File

Offset			Field	Format	Description
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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

Table 202. O2 (Optical Access) journal entries. QASY02JE/J4/J5 Field Description File (continued)					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		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

O3 (Optical Access) journal entries

This table provides the format of the O3 (Optical Access) journal entries.

Table 203. O3 (Optical Access) journal entries. QASY03JE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630, "Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 for the field listing.
156	224	610	Entry Type	Char(1)	**A** Change Volume Attributes **B** Backup Volume **C** Convert Backup Volume to Primary **E** Export **I** Initialize **K** Check Volume **L** Change Authorization List **M** Import **N** Rename **R** Absolute Read
157	225	611	Device Name	Char(10)	Library LUD name
167	235	621	CSI Name	Char(8)	Side Object Name

Table 203. O3 (Optical Access) journal entries. QASY03JE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
175	243	629	CSI Library	Char(10)	Side Object Library
185	253	639	Old Volume Name	Char(32)	Old Optical volume 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 Backup, Convert, Initialize, and Rename.

2
Used for Import, Export, and Change Authorization List only.

3
Used for Change Authorization List only.

4
Used for Sector Read only.

PA (Program Adopt) journal entries

This table provides the format of the PA (Program Adopt) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_PA table function: AUDIT_JOURNAL_PA

Table 204. PA (Program Adopt) journal entries. QASYPAJE/J4/J5 Field Description File

Offset			Field	Format	Description
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 for field listing.

Table 204. PA (Program Adopt) journal entries. QASYPAJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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** Change object's SETUID, SETGID, or Restricted rename and unlink 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		Reserved	Char(18)	
		649	ISVTX mode	Char(1)	The current restricted rename and unlink (ISVTX) mode indicator. **Y** The ISVTX mode indicator is on for the object. **N** The ISVTX mode indicator is not on for the object.
		650	Previous ISVTX mode	Char(1)	The previous restricted rename and unlink (ISVTX) mode indicator. **Y** The ISVTX mode indicator was on for the object. **N** The ISVTX mode indicator was not on for the object.
		651	Previous SETUID Mode	Char(1)	The previous Set effective user ID (SETUID) mode indicator. **Y** The SETUID mode bit was on for the object. **N** The SETUID mode bit was not on for the object.

Table 204. PA (Program Adopt) journal entries. QASYPAJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		652	Previous SETGID Mode	Char(1)	The previous Set effective group ID (SETGID) mode indicator. **Y** The SETGID mode bit was on for the object. **N** The SETGID mode bit was not on for the object.
		653	Reserved	Char(14)	
	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 current Set effective user ID (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 current Set effective group ID (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.

Table 204. PA (Program Adopt) journal entries. QASYPAJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
	882	1268	Path Name CCSID	Binary(5)	The coded character set identifier for the path name.
	886	1272	Path Name Country or Region ID	Char(2)	The Country or Region ID for the path name.
	888	1274	Path Name Language ID	Char(3)	The language ID for the path name.
	891	1277	Path Name Length	Binary(4)	The length of the path name.
	893	1279	Path Name Indicator	Char(1)	Path name indicator: **Y** The Path Name field contains complete absolute path name for the object. **N** The Path Name field does not contain an absolute path name for the object, instead it contains a relative path name. The Relative Directory File ID field is valid and can be used to form an absolute path name with this relative path name.
	894	1280	Relative Directory File ID[4]	Char(16)	When the Path Name Indicator field is N, this field contains the file ID of the directory that contains the object identified in the Path Name field. Otherwise it contains hex zeros.[4]
	910	1296	Path Name[5]	Char(5002)	The path name of the object.

1
These fields are used only for objects in the "root" (/), QOpenSys, 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 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 zeros.

4
If the Path Name Indicator field is N, but the Relative Directory File ID is hex zeros, then there was some error in determining the path name information.

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.

PF (PTF Operations) journal entries

This table provides the format of the PF (PTF Operations) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_PF table function: AUDIT_JOURNAL_PF

Table 205. PF (PTF Operations) journal entries. QASYPFJ5 Field Description File

Offset J5	Field	Format	Description
1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630 for field listing.
610	Entry Type	Char(1)	The type of entry. **P** PTF operations **L** PTF product(s) operation **I** PTF IPL operation

Table 205. PF (PTF Operations) journal entries. QASYPFJ5 Field Description File *(continued)*			
Offset			
J5	**Field**	**Format**	**Description**
611	Entry Action	Char(4)	The type of action. When entry type (J5 offset 610) is P this field can contain: **LOGF** PTF logged **LOAD** PTF loaded **SUPR** PTF superseded **TAPY** PTF temporarily applied **PAPY** PTF permanently applied **TRMV** PTF temporarily removed **PRMV** PTF permanently removed **DAMG** PTF damaged **PDLT** PTF deleted **EXTS** PTF exit program started **EXTE** PTF exit program ended When entry type (J5 offset 610) is L this field can contain: **REST** Product restored/installed **SAVE** Product saved **DELT** Product deleted **SYNC** User called QPZSYNC **GOPT** GO PTF option 7 or 8 invoked **INSP** INSPTF command invoked When entry type (J5 offset 610) is I this field can contain: **IPLU** Unattended IPL performed **IPLA** Attended IPL performed

Table 205. PF (PTF Operations) journal entries. QASYPFJ5 Field Description File *(continued)*			
Offset			
J5	**Field**	**Format**	**Description**
615	IPL Action for PTF	Char(4)	Action to take for PTF on the next IPL. This field will only contain data when entry type (J5 offset 610) is P and entry action (J5 offset 611) is not EXTS or EXTE. **NONE** No IPL action taken **ATMP** Apply temporarily at IPL **APRM** Apply permanently at IPL **RTMP** Remove temporarily at IPL **RPRM** Remove permanently at IPL **ATTP** Apply temporarily then permanently at IPL **RTTP** Remove temporarily then permanently at IPL
619	Product ID	Char(7)	Product ID or one of the values listed below. This field will only contain data when entry type (J5 offset 610) is P or L. ***ALL** All products ***FMW** Firmware ***LIST** List of products
626	Product VRM	Char(6)	Product version, release, modification in format vvrrmm or *ONLY. This field will only contain data when entry type (J5 offset 610) is P or L.
632	PTF ID	Char(7)	PTF identifier. This field will only contain data when entry type (J5 offset 610) is P.
639	Product Option	Char(4)	Product option or *ALL. This field will only contain data when entry type (J5 offset 610) is P or L and entry action (J5 offset 611) is not LOGF, PDLT, or SYNC.
643	Product Load	Char(4)	Product load identifier or *ALL. This field will only contain data when entry type (J5 offset 610) is P or L and entry action (J5 offset 611) is not LOGF, PDLT, or SYNC.
647	PTF Minimum Level	Char(2)	PTF minimum level. This field will only contain data when entry type (J5 offset 610) is P and entry action (J5 offset 611) is not LOGF or PDLT.
649	PTF Maximum Level	Char(2)	PTF maximum level. This field will only contain data when entry type (J5 offset 610) is P and entry action (J5 offset 611) is not LOGF or PDLT.

Table 205. PF (PTF Operations) journal entries. QASYPFJ5 Field Description File *(continued)*			
Offset			
J5	**Field**	**Format**	**Description**
651	Product Library	Char(10)	Product library or one of the values listed below. This field will only contain data when entry type (J5 offset 610) is P or L and entry action (J5 offset 611) is not LOGF or PDLT. ***ALL** All product libraries ***FMW** Firmware
661	Action Pending	Char(1)	Action pending for PTF. This field will only contain data when entry type (J5 offset 610) is P and entry action (J5 offset 611) is not EXTS or EXTE. **N** No action pending for PTF **Y** Action pending for PTF
662	Superseded-by PTF	Char(7)	Superseded-by PTF ID. This field will only contain data when entry type (J5 offset 610) is P and entry action (J5 offset 611) is SUPR.
669	PTF Exit Program	Char(10)	PTF exit program name. This field will only contain data when entry type (J5 offset 610) is P and entry action (J5 offset 611) is EXTS or EXTE.
679	PTF Exit Program Library	Char(10)	PTF exit program library name. This field will only contain data when entry type (J5 offset 610) is P and entry action (J5 offset 611) is EXTS or EXTE.
689	PTF Exit Action	Char(1)	PTF exit program action. This field will only contain data when entry type (J5 offset 610) is P and entry action (J5 offset 611) is EXTS or EXTE. **0** Remove temporarily **1** Apply temporarily **2** Apply permanently **3** Remove permanently **4** Pre-remove temporarily **5** Pre-apply temporarily **6** Pre-apply permanently **7** Pre-remove permanently

Table 205. PF (PTF Operations) journal entries. QASYPFJ5 Field Description File *(continued)*			
Offset			
J5	**Field**	**Format**	**Description**
690	QPZSYNC Parameter One	Char(1)	QPZSYNC function first parameter. This field will only contain data when entry type (J5 offset 610) is L and entry action (J5 offset 611) is SYNC.
691	Install Apply Type	Char(10)	PTF install apply type. This field will only contain data when entry type (J5 offset 610) is L and entry action (J5 offset 611) is GOPT or INSP. ***DLYIPL** Mark PTFs for delayed apply and IPL ***DLYALL** Mark PTFs for delayed apply ***IMMDLY** Apply immediate PTFs and mark delayed PTFS for delayed apply ***IMMONLY** Only apply immediate PTFs
701	Device Name	Char(10)	PTF install device name or one of the values listed below. This field will only contain data when entry type (J5 offset 610) is L and entry action (J5 offset 611) is GOPT or INSP. ***SERVICE** Install PTFs received from service support system ***NONE** No PTFs are loaded, PTFs already loaded are applied
711	Image Catalog	Char(10)	PTF install image catalog name or one of the values listed below. This field will only contain data when entry type (J5 offset 610) is L and entry action (J5 offset 611) is GOPT or INSP. ***NONE** No image catalog ***NETOPT** Network Optical ***RMTDEV** Remote device
721	Prompt for Media	Char(10)	PTF install prompt for media. This field will only contain data when entry type (J5 offset 610) is L and entry action (J5 offset 611) is GOPT or INSP. ***SNGVOLSET** Prompt for volumes in single volume set ***MLTVOLSET** Prompt for volumes in multiple volume sets ***MLTSRV** Prompt for volumes in multiple volume sets then load from *SERVICE

Table 205. PF (PTF Operations) journal entries. QASYPFJ5 Field Description File *(continued)*			
Offset			
J5	**Field**	**Format**	**Description**
731	Copy PTFs	Char(1)	Copy PTF save files and cover letters into *SERVICE on PTF install. This field will only contain data when entry type (J5 offset 610) is L and entry action (J5 offset 611) is GOPT or INSP. **N** PTFs not copied **Y** PTFs copied
732	Omit PTFs	Char(1)	PTFs omitted on PTF install. This field will only contain data when entry type (J5 offset 610) is L and entry action (J5 offset 611) is GOPT or INSP. **N** PTFs not omitted **Y** PTFs omitted
733	Automatic IPL	Char(1)	Automatic IPL on PTF install. This field will only contain data when entry type (J5 offset 610) is L and entry action (J5 offset 611) is GOPT or INSP. **N** No automatic IPL **Y** Automatic IPL performed
734	IPL Restart Type	Char(5)	IPL restart type for automatic IPL on PTF install. This field will only contain data when entry type (J5 offset 610) is L and entry action (J5 offset 611) is GOPT or INSP. ***SYS** System determines how much to restart ***FULL** All parts of system, including hardware, are restarted **IPLA** IPL attributes
739	HIPER Only PTFs	Char(1)	Only HIPER PTFs loaded on PTF install. This field will only contain data when entry type (J5 offset 610) is L and entry action (J5 offset 611) is GOPT or INSP. **N** All PTFs loaded. **Y** Only HIPER PTFs loaded

Table 205. PF (PTF Operations) journal entries. QASYPFJ5 Field Description File *(continued)*			
Offset J5	**Field**	**Format**	**Description**
740	IPL Type	Char(1)	IPL type. This field will only contain data when entry type (J5 offset 610) is I. **0** Unattended IPL **1** Attended IPL **2** IPL during operating system install
741	Abnormal IPL	Char(1)	Abnormal IPL. This field will only contain data when entry type (J5 offset 610) is I. **N** Normal IPL **Y** Abnormal IPL
742	LIC Restored	Char(1)	LIC restored during this IPL. This field will only contain data when entry type (J5 offset 610) is I. **N** LIC not restored **Y** LIC restored
743	Restart SAG	Char(1)	Restart Shared Activation Group (SAG) during IPL after applying PTFs. This field will only contain data when entry type (J5 offset 610) is I. **N** SAG not restarted **Y** SAG restarted
744	Re-IPL LIC	Char(1)	Re-IPL of LIC requested during IPL. This field will only contain data when entry type (J5 offset 610) is I. **N** No re-IPL of LIC requested **Y** Re-IPL of LIC requested

PG (Primary Group Change) journal entries

This table provides the format of the PG (Primary Group Change) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_PG table function: AUDIT_JOURNAL_PG

Table 206. PG (Primary Group Change) journal entries. QASYPGJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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)	
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

Table 206. PG (Primary Group Change) journal entries. QASYPGJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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		(Reserved Area)	Char(20)	
					Previous authorities
		692	Object Existence	Char(1)	**Y** *OBJEXIST
		693	Object Management	Char(1)	**Y** *OBJMGT
		694	Object Operational	Char(1)	**Y** *OBJOPR
		695	Object Alter	Char(1)	**Y** *OBJALTER
		696	Object Reference	Char(1)	**Y** *OBJREF
		697	Authorization List Management	Char(1)	**Y** *AUTLMGT
		698	Read Authority	Char(1)	**Y** *READ
		699	Add Authority	Char(1)	**Y** *ADD
		700	Update Authority	Char(1)	**Y** *UPD

Table 206. PG (Primary Group Change) journal entries. QASYPGJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		701	Delete Authority	Char(1)	**Y** *DLT
		702	Execute Authority	Char(1)	**Y** *EXECUTE
		703	Exclude Authority	Char(1)	**Y** *EXCLUDE
		704	(Reserved Area)	Char(8)	
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.

Table 206. PG (Primary Group Change) journal entries. QASYPGJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
	1035	1422	Path Name CCSID	Binary(5)	The coded character set identifier for the path name.
	1040	1426	Path Name Country or Region ID	Char(2)	The Country or Region ID for the path name.
	1042	1428	Path Name Language ID	Char(3)	The language ID for the path name.
	1045	1431	Path Name Length	Binary(4)	The length of the path name.
	1047	1433	Path Name Indicator	Char(1)	Path name indicator: **Y** The Path Name field contains complete absolute path name for the object. **N** The Path Name field does not contain an absolute path name for the object, instead it contains a relative path name. The Relative Directory File ID field is valid and can be used to form an absolute path name with this relative path name.
	1048	1434	Relative Directory File ID[3]	Char(16)	When the Path Name Indicator field is N, this field contains the file ID of the directory that contains the object identified in the Path Name field. Otherwise it contains hex zeros.[3]
	1064	1450	Path Name[4]	Char(5002)	The path name of the object.

1
These fields are used only for objects in the "root" (/), QOpenSys, 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
If the Path Name Indicator field is N, but the Relative Directory File ID is hex zeros, then there was some error in determining the path name information.

4
This is a variable length field. The first 2 bytes contain the length of the path name.

5
A value of *N implies that 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.

PO (Printer Output) journal entries

This table provides the format of the PO (Printer Output) journal entries.

Table 207. PO (Printer Output) journal entries. QASYPOJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**`J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630, "Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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 **R** Ready (Set by QSPSETWI API) **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]
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].

Table 207. PO (Printer Output) journal entries. QASYPOJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**`J4**	**J5**			
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.
		1143	Spooled File Job system Name	Char (8)	The name of the system on which the spooled file resides.
		1151	Spooled File Create Date	Char (7)	The spooled file create date (CYYMMDD)
		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
		1179	Output Queue ASP Name	Char(10)	ASP name for output queue library.
		1189	Output Queue ASP Number	Char(5)	ASP number for output queue library.
		1194	Spooled File Create Date UTC	Char(7)	The spooled file create date in UTC (This is the same date as the Spool File Create Date (offset 1151) only in UTC).

Table 207. PO (Printer Output) journal entries. QASYPOJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**`J4**	**J5**			
		1201	Spooled File Create Time UTC	Char(6)	The spooled file create time in UTC (This is the same time as the Spool File Create Time (offset 1158) only in UTC)

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.

PS (Profile Swap) journal entries

This table provides the format of the PS (Profile Swap) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_PS table function: AUDIT_JOURNAL_PS

Table 208. PS (Profile Swap) journal entries. QASYPSJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See “Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)” on page 630,“Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)” on page 632, and “Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)” on page 633 for field listing.

Table 208. PS (Profile Swap) journal entries. QASYPSJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
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 **T** Telnet QIBM_QTG_DEVINIT exit program profile swap. **U** Telnet QIBM_QTG_DEVINIT exit program profile override. **V** User profile authenticated
157	225	611	User Profile	Char(10)	User profile name. When entry type (J5 offset 610) is T this is the name returned by the exit program. When entry type (J5 offset 610) is U this is the name used by the exit program.
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. When entry type (J5 offset 610) is T this is the name negotiated by the client or blanks if no user was negotiated.

Table 208. PS (Profile Swap) journal entries. QASYPSJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
185	253	639	New Target User Profile	Char(10)	New pass-through target user profile. When entry type (J5 offset 610) is T this is the name returned by the exit program. This is the same value as returned in the User Profile (J5 offset 611) field.
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 that the profile token is valid.

PU (PTF Object Change) journal entries

This table provides the format of the PU (PTF Object Change) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_PU table function: AUDIT_JOURNAL_PU

Table 209. PU (PTF Object Change) journal entries. QASYPUJ5 Field Description File			
Offset	**Field**	**Format**	**Description**
J5			
1			Heading fields common to all entry types. See “Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)” on page 630 for field listing.
610	Entry Type	Char(1)	The type of entry. **L** Library PTF object **D** Directory PTF object **S** LIC PTF object

Table 209. PU (PTF Object Change) journal entries. QASYPUJ5 Field Description File *(continued)*			
Offset			
J5	**Field**	**Format**	**Description**
611	Entry Action	Char(1)	The type of action. **C** Changed PTF object **N** New PTF object
612	PTF Operation	Char(1)	The PTF operation. **A** Apply **R** Remove
613	Product ID	Char(7)	Product ID.
620	Product VRM	Char(6)	Product version, release, modification in format vvrrmm.
626	PTF ID	Char(7)	PTF identifier.
633	Product Option	Char(4)	Product option.
637	Product Load	Char(4)	Product load identifier.
641	Product Minimum Level	Char(2)	Product minimum level.
643	Product Maximum Level	Char(2)	Product maximum level.
645	Product Library	Char(10)	Product library.
655	Object Name[6]	Char(10)	Object name.
665	Object Library[6]	Char(10)	Object library.
675	Object Type[6]	Char(7)	Object type.
682	RU Name[7]	Char(8)	Replaceable Unit (RU) name.
690	(Reserved Area)	Char(58)	
748	Object Name Length [1,8]	Binary(4)	The length of the object name.
750	Object Name CCSID[1,8]	Binary(5)	The coded character set identifier for the object name.
754	Object Name Country or Region ID[1,8]	Char(2)	The Country or Region ID for the object name.
756	Object Name Language ID[1,8]	Char(3)	The language ID for the object name.
759	(Reserved area)	Char(3)	
762	Parent File ID[1,2,8]	Char(16)	The file ID of the parent directory.
778	Object File ID[1,2,8]	Char(16)	The file ID of the object.
794	Object Name[1,8]	Char(512)	The name of the object.

Table 209. PU (PTF Object Change) journal entries. QASYPUJ5 Field Description File *(continued)*			
Offset			
J5	**Field**	**Format**	**Description**
1306	Object File ID[8]	Char(16)	The file ID of the object.
1322	ASP Name[5]	Char(10)	The name of the ASP device.
1332	ASP Number[5]	Char(5)	The number of the ASP device.
1337	Path Name CCSID[8]	Binary(5)	The coded character set identifier for the path name.
1341	Path Name Country or Region ID[8]	Char(2)	The Country or Region ID for the path name.
1343	Path Name Language ID[8]	Char(3)	The language ID for the path name.
1346	Path Name Length[8]	Binary(4)	The length of the path name.
1348	Path Name Indicator[8]	Char(1)	Path name indicator: **Y** The Path Name field contains complete absolute path name for the object. **N** The Path Name field does not contain an absolute path name for the object, instead it contains a relative path name. The Relative Directory File ID field is valid and can be used to form an absolute path name with this relative path name.
1349	Relative Directory File ID[3,8]	Char(16)	When the Path Name Indicator field is N, this field contains the file ID of the directory that contains the object identified in the Path Name field. Otherwise it contains hex zeros.[3]
1365	Path Name[4,8]	Char(5002)	The path name of the object.

Table 209. PU (PTF Object Change) journal entries. QASYPUJ5 Field Description File *(continued)*			
Offset			
J5	**Field**	**Format**	**Description**
1 These fields are used only for objects in the "root" (/), QOpenSys, 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** If the Path Name Indicator field is N, but the Relative Directory File ID is hex zeros, then there was some error in determining the path name information. **4** This is a variable length field. The first two bytes contain the length of the path name. **5** This field will contain blanks when entry type (J5 offset 610) is L or S. Library PTF objects, entry type L, will always be in *SYSBAS. **6** This field will only contain data when entry type (J5 offset 610) is L. **7** This field will only contain data when entry type (J5 offset 610) is S. **8** This field will only contain data when entry type (J5 offset 610) is D.			

PW (Password) journal entries

This table provides the format of the PW (Password) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_PW table function: AUDIT_JOURNAL_PW

Table 210. 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 “Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)” on page 630,“Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)” on page 632, and “Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)” on page 633 for field listing.

Table 210. PW (Password) journal entries. QASYPWJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
156	224	610	Violation Entry Type	Char(1)	The type of violation **A** APPC bind failure. **C** User authentication with the CHKPWD command failed. **D** Service tools user ID name not valid (QSYCHGDS API, CRTSSTUSR, CHGSSTUSR, DLTSSTUSR commands). **E** Service tools user ID password not valid (QSYCHGDS API, CRTSSTUSR, CHGSSTUSR, DLTSSTUSR commands). **P** Password not valid. **Q** Attempted signon (user authentication) failed because user profile is disabled. **R** Attempted signon (user authentication) failed because password was expired. This audit record might not occur for some user authentication mechanisms. Some authentication mechanisms do not check for expired passwords. **S** SQL Decryption password is not valid. **U** User name not valid. **X** Service tools user ID is disabled. **Y** Service tools user ID not valid (service tools interface). **Z** Service tools user ID password not valid (service tools interface).
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. When the entry type (J5 offset 610) is D, E, X, Y, or Z this field will contain the name of the interface being used.

Table 210. PW (Password) journal entries. QASYPWJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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.
		685[2]	Object Name	Char(10)	The name of the object being decrypted.
		695	Object Library	Char(10)	The library for the object being decrypted.
		705	Object Type	Char(8)	The type of object being decrypted.
		713	ASP Name[1]	Char(10)	The name of the ASP device.
		723	ASP Number[1]	Char(5)	The number of the ASP device.

1
If the object is in a library, this is the ASP information for the object's library. If the object is not in a library, this is the ASP information for the object.

2
If the object name is *N and the violation type is S, the user attempted to decrypt data in a host variable.

RA (Authority Change for Restored Object) journal entries

This table provides the format of the RA (Authority Change for Restored Object) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_RA table function: AUDIT_JOURNAL_RA

Table 211. RA (Authority Change for Restored Object) journal entries. QASYRAJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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 where the object is stored.

Table 211. RA (Authority Change for Restored Object) journal entries. QASYRAJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
177	245	631	Object Type	Char(8)	The type of object.
185	253	639	Restored Authorization List Name	Char(10)	The name of the authorization list on the restored object.
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		(Reserved Area)	Char(20)	
		652	Saved Authorization List Name	Char(10)	The name of the authorization list on the saved object.
		662	(Reserved Area)	Char(10)	
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.

Table 211. RA (Authority Change for Restored Object) journal entries. QASYRAJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
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 path name.
	980	1366	Path Name Country or Region ID	Char(2)	The Country or Region ID for the path name.
	982	1368	Path Name Language ID	Char(3)	The language ID for the path name.
	985	1371	Path Name Length	Binary(4)	The length of the path name.
	987	1373	Path Name Indicator	Char(1)	Path name indicator: **Y** The Path Name field contains complete absolute path name for the object. **N** The Path Name field does not contain an absolute path name for the object, instead it contains a relative path name. The Relative Directory File ID field is valid and can be used to form an absolute path name with this relative path name.
	988	1374	Relative Directory File ID[3]	Char(16)	When the Path Name Indicator field is N, this field contains the file ID of the directory that contains the object identified in the Path Name field. Otherwise it contains hex zeros.[3]
	1004	1390	Path Name[4]	Char(5002)	The path name of the object.

Table 211. RA (Authority Change for Restored Object) journal entries. QASYRAJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1 These fields are used only for objects in the "root" (/), QOpenSys, 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** If the Path Name Indicator field is N, but the Relative Directory File ID is hex zeros, then there was some error in determining the path name information. **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.					

RJ (Restoring Job Description) journal entries

This table provides the format of the RJ (Restoring Job Description) journal entries.

Table 212. RJ (Restoring Job Description) journal entries. QASYRJJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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 currently 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 212. RJ (Restoring Job Description) journal entries. QASYRJJE/J4/J5 Field Description File (continued)					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		664	Previous User Name	Char(10)	The name of the user profile previously specified in the job description.

RO (Ownership Change for Restored Object) journal entries

This table provides the format of the RO (Ownership Change for Restored Object) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_RO table function: AUDIT_JOURNAL_RO

Table 213. RO (Ownership Change for Restored Object) journal entries. QASYROJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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	Saved Owner	Char(10)	The name of the owner on the saved object.
195	263	649	Restored Owner	Char(10)	The name of the owner on the restored 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.

Table 213. RO (Ownership Change for Restored Object) journal entries. QASYROJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
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 path name.
	987	1373	Path Name Country or Region ID	Char(2)	The Country or Region ID for the path name.
	989	1375	Path Name Language ID	Char(3)	The language ID for the path name.
	992	1378	Path Name Length	Binary(4)	The length of the path name.
	994	1380	Path Name Indicator	Char(1)	Path name indicator: **Y** The Path Name field contains complete absolute path name for the object. **N** The Path Name field does not contain an absolute path name for the object, instead it contains a relative path name. The Relative Directory File ID field is valid and can be used to form an absolute path name with this relative path name.
	995	1381	Relative Directory File ID[3]	Char(16)	When the Path Name Indicator field is N, this field contains the file ID of the directory that contains the object identified in the Path Name field. Otherwise it contains hex zeros.[3]

Table 213. RO (Ownership Change for Restored Object) journal entries. QASYROJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	J4	J5			
	1011	1397	Path Name[4]	Char(5002)	The path name of the object.

1
These fields are used only for objects in the "root" (/), QOpenSys, 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
If the Path Name Indicator field is N, but the Relative Directory File ID is hex zeros, then there was some error in determining the path name information.

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.

RP (Restoring Programs that Adopt Authority) journal entries

This table provides the format of the RP (Restoring Programs that Adopt Authority) journal entries.

Table 214. RP (Restoring Programs that Adopt Authority) journal entries. QASYRPJE/J4/J5 Field Description File

Offset			Field	Format	Description
JE	J4	J5			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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 where 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.

Table 214. RP (Restoring Programs that Adopt Authority) journal entries. QASYRPJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
	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 path name.
	874	1260	Path Name Country or Region ID	Char(2)	The Country or Region ID for the path name.
	876	1262	Path Name Language ID	Char(3)	The language ID for the path name.
	879	1265	Path Name Length	Binary(4)	The length of the path name.
	881	1267	Path Name Indicator	Char(1)	Path name indicator: **Y** The Path Name field contains complete absolute path name for the object. **N** The Path Name field does not contain an absolute path name for the object, instead it contains a relative path name. The Relative Directory File ID field is valid and can be used to form an absolute path name with this relative path name.
	882	1268	Relative Directory File ID[3]	Char(16)	When the Path Name Indicator field is N, this field contains the file ID of the directory that contains the object identified in the Path Name field. Otherwise it contains hex zeros.[3]

Table 214. RP (Restoring Programs that Adopt Authority) journal entries. QASYRPJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	J4	J5			
	898	1284	Path Name[4]	Char(5002)	The path name of the object.

1
These fields are used only for objects in the "root" (/), QOpenSys, and user-defined 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
If the Path Name Indicator field is N, but the Relative Directory File ID is hex zeros, then there was some error in determining the path name information.

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.

RQ (Restoring Change Request Descriptor Object) journal entries

This table provides the format of the RQ (Restoring Change Request Descriptor Object) journal entries.

Table 215. RQ (Restoring Change Request Descriptor Object) journal entries. QASYRQJE/J4/J5 Field Description File

Offset			Field	Format	Description
JE	J4	J5			
1	1	1			Heading fields common to all entry types. See “Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)” on page 630,“Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)” on page 632, and “Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)” on page 633 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

RU (Restore Authority for User Profile) journal entries

This table provides the format of the RU (Restore Authority for User Profile) journal entries.

Table 216. RU (Restore Authority for User Profile) journal entries. QASYRUJE/J4/J5 Field Description File

Offset			Field	Format	Description
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **A** Restoring authority to user profiles
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

RZ (Primary Group Change for Restored Object) journal entries

This table provides the format of the RZ (Primary Group Change for Restored Object) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_RZ table function: AUDIT_JOURNAL_RZ

Table 217. RZ (Primary Group Change for Restored Object) journal entries. QASYRZJE/J4/J5 Field Description File

Offset			Field	Format	Description
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 for field listing.

Table 217. RZ (Primary Group Change for Restored Object) journal entries. QASYRZJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
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	Saved Primary Group	Char(10)	Primary group on the saved object.
195	263	649	Restored Primary Group	Char(10)	Primary group on the restored 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.
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.

Table 217. RZ (Primary Group Change for Restored Object) journal entries. QASYRZJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
	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 path name.
	987	1373	Path Name Country or Region ID	Char(2)	The Country or Region ID for the path name.
	989	1375	Path Name Language ID	Char(3)	The language ID for the path name.
	992	1378	Path Name Length	Binary(4)	The length of the path name.
	994	1380	Path Name Indicator	Char(1)	Path name indicator: **Y** The Path Name field contains complete absolute path name for the object. **N** The Path Name field does not contain an absolute path name for the object, instead it contains a relative path name. The Relative Directory File ID field is valid and can be used to form an absolute path name with this relative path name.
	995	1381	Relative Directory File ID[3]	Char(16)	When the Path Name Indicator field is N, this field contains the file ID of the directory that contains the object identified in the Path Name field. Otherwise it contains hex zeros.[3]
	1011	1397	Path Name[4]	Char(5002)	The path name of the object.

1
These fields are used only for objects in the "root" (/), QOpenSys, 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
If the Path Name Indicator field is N, but the Relative Directory File ID is hex zeros, then there was some error in determining the path name information.

4
This is a variable length field. The first 2 bytes contain the length of the path name.

SD (Change System Distribution Directory) journal entries

This table provides the format of the SD (Change System Distribution Directory) journal entries.

Table 218. SD (Change System Distribution Directory) journal entries. QASYSDJE/J4/J5 Field Description File

Offset			Field	Format	Description
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 for field listing.
156	224	610	Entry Type	Char(1)	The type of entry. **S** System directory change
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 **OWN** Change ownership **PRT** Print directory entry **RMV** Remove directory entry **RNM** Rename directory entry **RTV** Retrieve details **SUP** Supplier entry

Table 218. SD (Change System Distribution Directory) journal entries. QASYSDJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
160	228	614	Type of record	Char(4)	**DIRE** Directory **DPTD** Department details **DSTL** Distribution list **NICK** Nickname **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 **REFRSH** Refresh **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
		713	Nickname	Char(8)	Nickname being changed
		721	Nickname Old Owner	Char(10)	Nickname old owner name
		731	Nickname New Owner	Char(10)	Nickname new owner name
		741	Distribution list ID	Char(8)	Distribution list ID

Table 218. SD (Change System Distribution Directory) journal entries. QASYSDJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		749	Distribution list qualifier	Char(8)	Distribution list qualifier
		757	Distribution List Old Owner	Char(10)	Distribution list old owner name
		767	Distribution List New Owner	Char(10)	Distribution list new owner name

SE (Change of Subsystem Routing Entry) journal entries

This table provides the format of the SE (Change of Subsystem Routing Entry) journal entries.

Table 219. SE (Change of Subsystem Routing Entry) journal entries. QASYSEJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 for field listing.
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 where the object is stored.
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

Table 219. SE (Change of Subsystem Routing Entry) journal entries. QASYSEJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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

SF (Action to Spooled File) journal entries

This table provides the format of the SF (Action to Spooled File) journal entries.

Table 220. SF (Action to Spooled File) journal entries. QASYSFJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 for field listing.

Table 220. SF (Action to Spooled File) journal entries. QASYSFJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
156	224	610	Access Type	Char(1)	The type of entry **A** Spooled file read by someone other than the owner of the spooled file. **C** Spooled file created. **D** Spooled file deleted. **H** Spooled file held. **I** Create of inline file. **R** Spooled file released. **S** Spooled file saved. **T** Spooled file restored. **U** Security-relevant spooled file attributes changed. **V** Only non-security-relevant spooled file attributes changed. **X** Spooled file operation rejected by exit program.
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 (J5 offset 693) will be used.
219	287	673	Output Queue Name	Char(10)	The name of the output queue containing the spooled file.

Table 220. SF (Action to Spooled File) journal entries. QASYSFJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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 220. SF (Action to Spooled File) journal entries. QASYSFJE/J4/J5 Field Description File *(continued)*					
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.

Table 220. SF (Action to Spooled File) journal entries. QASYSFJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
	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.
	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.
		1089	Spooled File Job System Name	Char(8)	The name of the system on which the spooled file resides.
		1097	Spooled File Create Date	Char(7)	The spooled file create date (CYYMMDD).
		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
		1620	File ASP Name	Char(10)	ASP name for database file library.
		1630	File ASP Number	Char(5)	ASP number for database file library.
		1635	Output Queue ASP name	Char(10)	ASP name for output queue library.

Table 220. SF (Action to Spooled File) journal entries. QASYSFJE/J4/J5 Field Description File *(continued)*					
Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
		1645	Output Queue ASP number	Char(5)	ASP number for output queue library.
		1650	New Output Queue ASP Name	Char(10)	ASP name for new output queue library.
		1660	New Output Queue ASP Number	Char(5)	ASP number for new output queue library.
		1665	Old Spooled File Status	Char(3)	Old spooled file status.
		1668	New Spooled File Status	Char(3)	New spooled file status.
		1671	Original Creation Date	Char(7)	Original creation date.
		1678	Original Creation Time	Char(6)	Original creation time.
		1684	Old Spooled File Expiration Date	Char(7)	Old spooled file expiration date
		1691	New Spooled File Expiration Date	Char(7)	New spooled file expiration date
		1698	Spooled File Create Date UTC	Char(7)	The spooled file create date in UTC (This is the same date as the Spool File Create Date (offset 1097) only in UTC)
		1705	Spooled File Create Time UTC	Char(6)	The spooled file create time in UTC (This is the same time as the Spool File Create Time (offset 1104) only in UTC)
		1711	Registered security exit program	Char(10)	The name of the registered security exit program.
		1721	Registered security exit program library	Char(10)	The library name of the registered security exit program.
		1731	Registered security exit program ASP name	Char(10)	The ASP name of the registered security exit program.
		1741	Registered security exit program ASP number	Char(5)	The ASP number of the registered security exit program.

Table 220. SF (Action to Spooled File) journal entries. QASYSFJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1 This field is blank when the type of entry is I (inline print).					

SG (Asynchronous Signals) journal entries

This table provides the format of the SG (Asynchronous Signals) journal entries.

Table 221. SG (Asynchronous Signals) journal entries. QASYSGJ4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630 and "Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632 for field listing.
	224	610	Entry Type	Char(1)	The type of entry. **A** Asynchronous IBM i 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** End the process **U** End the request

Table 221. SG (Asynchronous Signals) journal entries. QASYSGJ4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	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 job'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 when the signal was generated. **Note:** The QWCCVTDT API can be used to convert a *DTS time stamp to other formats.

SK (Sockets Connections) journal entries

This table provides the format of the SK (Sockets Connections) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_SK table function: AUDIT_JOURNAL_SK

Table 222. SK (Sockets Connections) journal entries. QASYSKJ4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	1	1			Heading fields common to all entry types. See “Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)” on page 630 and “Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)” on page 632 for field listing.

Table 222. SK (Sockets Connections) journal entries. QASYSKJ4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
	224	610	Entry type	Char(1)	**A** Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_AD table function: AUDIT_JOURNAL_AD Accept **C** Connect **D** DHCP address assigned **F** Filtered mail **I** Inbound UDP traffic **O** Outbound UDP traffic **P** Port unavailable **R** Reject mail **S**[4] Successful secure connection **U** DHCP address not assigned **X** Failed System TLS connection
	225	611	Local IP Address[3]	Char(15)	The local IP address.
	240	626	Local port	Char(5)	The local port.
	245	631	Remote IP Address[3]	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 222. SK (Sockets Connections) journal entries. QASYSKJ4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	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.
		1570	Secure version	Char(10)	The security protocol including the specific version level, if available, used for the connection. The possible protocol prefixes include: TLS, DTLS, SSL, IKE, IPSEC, SSH. A specific example would be "TLSV1.2" if the connection is protected by System TLS using TLSv1.2. An entry for a non-operating system connection may contain a raw version value such as "0401" if the system inspection code encounters a version it doesn't understand.
		1580	Secure properties	CHAR(100)	The secure properties used for the connection. When entry type (J5 offset 610) is S this field varies based on the secure version field (J5 offset 1570). Where possible this field contains one or more space separated character strings describing the cryptographic algorithms and key sizes used for the connection. The algorithms and key sizes are presented in a character format associated with the secure version field. A TLSv1.2 entry may look like this: "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_ SHA384 ECDSA_SHA512 SECP521R1" An entry for a non-operating system connection may contain a protocol's internal algorithm representation values such as "C054 0703 29" if the system inspection code encounters unknown values. When entry type (J5 offset 610) is X this field contains a string that represents the TLS error code.

Table 222. SK (Sockets Connections) journal entries. QASYSKJ4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		1680	Secure information	Char(100)	Additional information for the secure connection. When entry type (J5 offset 610) is X this field contains a string that describes the failure. When entry type (J5 offset 610) is S this field may contain additional attributes for the secure connection. For example, for IPSEC connections it contains the VPN Connection Name.

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 that the DHCP server assigned to the requesting client.

3 These fields only support IPv4 addresses.

4 When entry type is S, secure connection means a secure protocol was used, not that the algorithms used are considered secure. A system operator needs to review the secure version field and the secure properties field to determine the level of security.

SM (Systems Management Change) journal entries

This table provides the format of the SM (Systems Management Change) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_SM table function: AUDIT_JOURNAL_SM.

Table 223. SM (Systems Management Change) journal entries. QASYSMJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See “Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)” on page 630,“Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)” on page 632, and “Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)” on page 633 for field listing.

Table 223. SM (Systems Management Change) journal entries. QASYSMJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
156	224	610	Entry Type	Char(1)	Function accessed **B** Backup list changed **C** Automatic cleanup options **D** DRDA **F** HFS file system **M** Change DDM TCP/IP Attributes (CHGDDMTCPA) CL command **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)	When Entry Type (J5 offset 610) is S this field contains the sequence number of the action.
162	230	616	Message ID	Char(7)	When Entry Type (J5 offset 610) is S this field contains the message ID associated with the action.

Table 223. SM (Systems Management Change) journal entries. QASYSMJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
169	237	623	Relational Database Name	Char(18)	When Entry Type (J5 offset 610) is D or M this field contains the name of the relational database (RDB). When Access Type(J5 offset 611) is R this field may contain one of the special values: ***ALL** All entries in the RDB directory were removed. ***ALLRMT** All entries except the *LOCAL entry in the RDB directory were removed.
187	255	641	File System Name	Char(10)	When Entry Type (J5 offset 610) is F this field contains the name of the file system.
197	265	651	Backup Option Changed	Char(10)	When Entry Type (J5 offset 610) is O this field contains the backup option that was changed.
207	275	661	Backup List Change	Char(10)	When Entry Type (J5 offset 610) is B this field contains the name of the backup list that was changed.
217	285	671	Network File Name	Char(10)	When Entry Type (J5 offset 610) is N this field contains the name of the network file that was used.
227	295	681	Network File Member	Char(10)	When Entry Type (J5 offset 610) is N this field contains the name of the member of the network file.
237	305	691	Network File Number	Zoned(6,0)	When Entry Type (J5 offset 610) is N this field contains the number of the network file.
243	311	697	Network File Owner	Char(10)	When Entry Type (J5 offset 610) is N this field contains the name of the user profile that owns the network file.
253	321	707	Network File Originating User	Char(8)	When Entry Type (J5 offset 610) is N this field contains the name of the user profile that originated the network file.
261	329	715	Network File Originating Address	Char(8)	When Entry Type (J5 offset 610) is N this field contains the address that originated the network file.
		723[1]	RDB Alias	Char(18)	When Entry Type (J5 offset 610) is D this field contains the RDB alias name. This value may be *NONE.

Table 223. SM (Systems Management Change) journal entries. QASYSMJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		741	Remote Location	Char(254)	When Entry Type (J5 offset 610) is D this field contains the remote location name of the system on which the RDB is located, if available. **remote-location-name** The remote location name is in one of the following formats: • SNA remote location name (LU name). • SNA remote network identifier and remote location name separated by a period. • IPv4 address in dotted decimal form. • IPv6 address in colon hexadecimal form. • IP host domain name. ***ARDPGM** The RDB is accessed by using the Application Requester Driver (ARD) program. ***LOCAL** The system database on this system. ***LOOPBACK** This value is an alias for the IP address of the host system. ***MIRROR** The RDB is accessed on the other system for a Db2 Mirror relationship.
		995	Previous Remote Location	Char(254)	When Entry Type (J5 offset 610) is D this field contains the previous Remote Location value, if available.
		1249	Remote Location Type	Char(10)	When Entry Type (J5 offset 610) is D this field contains the remote location type, if available. ***IP** The RDB is found using a host name or an internet address over a TCP/IP connection. ***SNA** The RDB is accessed using a Systems Network Architecture (SNA) address and protocol.
		1259	Previous Remote Location Type	Char(10)	When Entry Type (J5 offset 610) is D this field contains the previous Remote Location Type value, if available.
		1269	Remote Port or Service	Char(14)	When Entry Type (J5 offset 610) is D and Remote Location Type (J5 offset 1249) is *IP this field contains the relational database entry port number or service name that is used at the remote location to communicate with the system on which the RDB is located, if available.

Table 223. SM (Systems Management Change) journal entries. QASYSMJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		1283	Previous Remote Port or Service	Char(14)	When Entry Type (J5 offset 610) is D and Remote Location Type (J5 offset 1249) is *IP this field contains the previous Remote Port or Service value, if available.
		1297	Preferred Authenticatio n	Char(10)	When Entry Type (J5 offset 610) is D this field contains the preferred authentication method on a connection request, if available. ***ENCRYPTED** User ID and encrypted password. ***ENCUSRPWD** Encrypted user ID and encrypted password. ***KERBEROS** Authentication occurs using Kerberos. ***USRENCPWD** User ID and encrypted password. ***USRID** User ID only. ***USRIDPWD** User ID and password.
		1307	Previous Preferred Authenticatio n	Char(10)	When Entry Type (J5 offset 610) is D this field contains the previous Preferred Authentication value, if available.
		1317	Lower Authenticatio n	Char(11)	When Entry Type (J5 offset 610) is D this field indicates whether an authentication method lower than what was specified for the preferred method will be accepted during negotiation with the server, if available. ***ALWLOWER** Allow negotiation of a lower authentication method. ***NOALWLOWER** Do not allow negotiation of a lower authentication method.
		1328	Previous Lower Authenticatio n	Char(11)	When Entry Type (J5 offset 610) is D this field contains the previous Lower Authentication value, if available.

Table 223. SM (Systems Management Change) journal entries. QASYSMJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		1339	Encryption Algorithm	Char(5)	When Entry Type (J5 offset 610) is D and Remote Location Type (J5 offset 1249) is *IP this field contains the encryption algorithm to be used initially on the connection request, if available. ***AES** Advanced Encryption Standard (AES) is to be initially used. ***DES** Data Encryption Standard (DES) is to be initially used.
		1344	Previous Encryption Algorithm	Char(5)	When Entry Type (J5 offset 610) is D and Remote Location Type (J5 offset 1249) is *IP this field contains the previous Encryption Algorithm value, if available.
		1349	Secure Connection	Char(5)	When Entry Type (J5 offset 610) is D and Remote Location Type (J5 offset 1249) is *IP this field indicates whether Transport Layer Security (TLS) is to be used on a DDM/DRDA TCP/IP connection request, if available. ***NONE** TLS is not used. ***SSL** TLS is used (means the same as *TLS). ***TLS** TLS is used.
		1354	Previous Secure Connection	Char(5)	When Entry Type (J5 offset 610) is D and Remote Location Type (J5 offset 1249) is *IP this field contains the previous Secure Connection value, if available.
		1359	APPC Device Description	Char(10)	When Entry Type (J5 offset 610) is D and Remote Location Type (J5 offset 1249) is *SNA this field contains the Advanced Program-to-Program Communications (APPC) device description on this system that is used with this RDB entry, if available. This field may contain the special value: ***LOC** If APPC is being used, the system determines which device description is used.
		1369	Previous APPC Device Description	Char(10)	When Entry Type (J5 offset 610) is D and Remote Location Type (J5 offset 1249) is *SNA this field contains the previous APPC Device Description value, if available.

Table 223. SM (Systems Management Change) journal entries. QASYSMJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		1379	Local Location	Char(8)	When Entry Type (J5 offset 610) is D and Remote Location Type (J5 offset 1249) is *SNA this field contains the local location name by which this system is identified to the system on which the RDB is located, if available. This field may contain one of the special values: ***LOC** If APPC is being used, the system determines which local location name is used. ***NETATR** The LCLLOCNAME value specified in the system network attributes is used.
		1387	Previous Local Location	Char(8)	When Entry Type (J5 offset 610) is D and Remote Location Type (J5 offset 1249) is *SNA this field contains the previous Local Location value, if available.
		1395	Remote Network ID	Char(8)	When Entry Type (J5 offset 610) is D and Remote Location Type (J5 offset 1249) is *SNA this field contains the remote network identifier of the system on which the RDB is located, if available. This field may contain one of the special values: ***LOC** If APPC is being used, the system determines which remote network identifier is used. ***NETATR** The remote network identifier specified in the network attributes is used. ***NONE** No remote network identifier is used.
		1403	Previous Remote Network ID	Char(8)	When Entry Type (J5 offset 610) is D and Remote Location Type (J5 offset 1249) is *SNA this field contains the previous Remote Network ID value, if available.

Table 223. SM (Systems Management Change) journal entries. QASYSMJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		1411	Remote Mode	Char(8)	When Entry Type (J5 offset 610) is D and Remote Location Type (J5 offset 1249) is *SNA this field contains the mode name to use with the remote location name to communicate with the system on which the RDB is located, if available. This field may contain one of the special values: ***NETATR** The mode in the network attributes is used. **BLANK** A mode name of all blanks is used.
		1419	Previous Remote Mode	Char(8)	When Entry Type (J5 offset 610) is D and Remote Location Type (J5 offset 1249) is *SNA C this field contains the previous Remote Mode value, if available.
		1427	Transaction Program	Char(19)	When Entry Type (J5 offset 610) is D and Remote Location Type (J5 offset 1249) is *SNA this field contains the name of the transaction program to use with RDB entry, if available. **transaction-program-name** The transaction program name is in one of the following formats: • A 4-byte hexadecimal name. For example, X'07F6C4C2'. • An 8-byte character name.
		1446	Previous Transaction Program	Char(19)	When Entry Type (J5 offset 610) is D and Remote Location Type (J5 offset 1249) is *SNA this field contains the previous Transaction Program value, if available.
		1465	ARD Library	Char(10)	When Entry Type (J5 offset 610) is D, Remote Location (J5 offset 741) is *ARDPGM, and ARD Program is not *DRDA this field contains the library containing the Application Requester Driver (ARD) program, if available. This value may be *LIBL or *CURLIB.
		1475	Previous ARD Library	Char(10)	When Entry Type (J5 offset 610) is D, Remote Location (J5 offset 741) is *ARDPGM, and ARD Program is not *DRDA this field contains the previous ARD Library value, if available.
		1485	ARD Program	Char(10)	When Entry Type (J5 offset 610) is D and Remote Location (J5 offset 741) is *ARDPGM this field contains the ARD program to be called to process SQL requests directed to the RDB, if available.

Table 223. SM (Systems Management Change) journal entries. QASYSMJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		1495	Previous ARD Program	Char(10)	When Entry Type (J5 offset 610) is D and Remote Location (J5 offset 741) is *ARDPGM this field contains the previous ARD Program value, if available.
		1505	Autostart Server	Char(5)	When Entry Type (J5 offset 610) is M this field indicates whether the DDM server is automatically started. This value was set using the Change DDM TCP/IP Attributes (CHGDDMTCPA) CL command, if available. ***YES** Automatically start the DDM server. ***NO** Do not automatically start the DDM server.
		1510	Previous Autostart Server	Char(5)	When Entry Type (J5 offset 610) is M this field contains the previous Autostart Server value, if available.
		1515	Lowest Authenticatio n Method	Char(10)	When Entry Type (J5 offset 610) is D and Access Type (J5 offset 611) is C, D, or S, or when Entry Type is D and Remote location (J5 offset 741) is *LOCAL, or when Entry Type is M this field contains the lowest level of password security required, if available. This value was set using the Change DDM TCP/IP Attributes (CHGDDMTCPA) CL command. ***ENCRYPTED** User ID and encrypted password. Same as *USRENCPWD. ***ENCUSRPWD** Encrypted user ID and encrypted password. ***KERBEROS** Authentication occurs using Kerberos. ***NO** User ID only. Same as *USRID. ***USRID** User ID only. ***USRENCPWD** User ID and encrypted password. ***USRIDPWD** User ID and password. ***VLDONLY** User ID only but if a password is sent on the request, it must be valid. ***YES** User ID and password. Same as *USRIDPWD.

Table 223. SM (Systems Management Change) journal entries. QASYSMJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		1525	Previous Lowest Authentication Method	Char(10)	When Entry Type (J5 offset 610) is M this field contains the previous Lowest Authentication Method value, if available.
		1535	Lowest Encryption Algorithm	Char(5)	When Entry Type (J5 offset 610) is D and Access Type (J5 offset 611) is A, C, R, or T, or when Entry Type is D and Remote location (J5 offset 741) is *LOCAL, or when Entry Type is M this field contains the lowest encryption algorithm allowed on an incoming connection request, if available. This value was set using the Change DDM TCP/IP Attributes (CHGDDMTCPA) CL command. ***AES** Advanced Encryption Standard (AES) allowed. ***DES** Data Encryption Standard (DES) or higher encryption algorithm allowed.
		1540	Previous Lowest Encryption Algorithm	Char(5)	When Entry Type (J5 offset 610) is M this field contains the previous Lowest Encryption Algorithm value, if available.

1
Fields starting at offset 723 are not available in the QASYSMJ5 model outfile. The information in these fields is in the audit journal entry and can be queried with the SYSTOOLS.AUDIT_JOURNAL_SM table function.

SO (Server Security User Information Actions) journal entries

This table provides the format of the SO (Server Security User Information Actions) journal entries.

Table 224. SO (Server Security User Information Actions) journal entries. QASYSOJE/J4/J5 Field Description File

Offset			Field	Format	Description
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 for field listing.

Table 224. SO (Server Security User Information Actions) journal entries. QASYSOJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
156	224	610	Entry Type	Char(1)	The type of entry **A** Add entry **C** Change entry **R** Remove entry **T** Retrieve entry
157	225	611	User Profile	Char(10)	The name of the user profile.
	235	621	User Information Entry Type	Char(1)	**N** Entry type not specified. **U** Entry is a user application information entry. **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.

ST (Service Tools Action) journal entries

This table provides the format of the ST (Service Tools Action) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_ST table function: AUDIT_JOURNAL_ST

Table 225. ST (Service Tools Action) journal entries. QASYSTJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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. **AN** ANZJVM **AR** ARM diagnostic trace (see ARMSRV QShell command) **AS** Storage altered by Display/Alter/Dump service tool or by a remote service tool debugger **CD** QTACTLDV, QTADMPDV **CE** QWTCTLTR **CS** STRCPYSCN **CT** DMPCLUTRC **DC** DLTCMNTRC **DD** DMPDLO **DF** QWTDMPFR, QWTDMPLF **DI** QSCDIRD **DJ** DMPJVM, QPYRTJVM **DM** DMPMEMINF **DO** DMPOBJ

Table 225. ST (Service Tools Action) journal entries. QASYSTJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
					DS DMPSYSOBJ, QTADMPTS, QTADMPDV, QWTDMPLF **DU** DMPUSRPRF **DW** STRDW, ENDDW, ADDDWDFN, RMVDWDFN **EC** ENDCMNTRC **ER** ENDRMTSPT **FF** FFDC (First Failure Data Capture) **GS** QSMGSSTD **HD** QYHCHCOP (DASD) **HL** QYHCHCOP (LPAR)
					JW STRJW, ENDJW, ADDJWDFN, RMVJWDFN **LC** EPT created **LD** EPT deleted **LE** EPT for the job has been changed **LF** System EPT has been fixed up **LG** Entries in the EPT have been changed **LH** EPT compared

Table 225. ST (Service Tools Action) journal entries. QASYSTJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
					LI EPT entries displayed **MC** QWTMAINT (change) **MD** QWTMAINT (dump) **MP** End system job **MQ** Restart system job **OP** Operations console **PC** PRTCMNTRC
					PE PRTERRLOG, QTADMPDV **PI** PRTINTDTA, QTADMPDV **PS** QP0FPTOS **SC** STRCMNTRC, QSCCHGCT **SE** QWTSETTR
					SF QWCCDSIC, QWVRCSTK (Display internal stack entry) **SJ** STRSRVJOB **SN** QPZSYNC **SR** STRRMTSPT **SS** QFPHPSF **ST** STRSST **SV** QSRSRV **TA** TRCTCPAPP

Table 225. ST (Service Tools Action) journal entries. QASYSTJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
					TC TRCCNN (*FORMAT specified) **TE** ENDTRC, ENDPEX, TRCJOB(*OFF or *END specified) **TI** TRCINT, or TRCCNN with SET(*ON), SET(*OFF), or SET(*END) **TO** QTOBSRV **TQ** QWCTMQTM **TS** STRTRC, STRPEX, TRCJOB(*ON specified)
					UD QTAUPDDV **WE** ENDWCH, QSCEWCH **WS** STRWCH, QSCSWCH **WT** WRKTRC **WW** WRKWCH, QSCRWCHI, QSCRWCHL
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.
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	LIC RU Name[11]	Char(8)	LIC RU name.
301	369	755	Folder Path[8]	Char(63)	The folder containing the document library object

Table 225. ST (Service Tools Action) journal entries. QASYSTJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	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. **A**[6] Activate **D**[6] Deactivate **Y**[7] Collection of trace information started **N**[7] Collection of trace information stopped and trace information written to spooled file **E**[7] 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

Table 225. ST (Service Tools Action) journal entries. QASYSTJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		907	Console Type [4]	Char(10)	The console type. Possible values are: • *DIRECT • *LAN • *HMC
		917	Console action [4]	Char(10)	The console action. Possible values are: • *RECOVERY • *TAKEOVER
		927	Address family [4]	Char(10)	The address family. • *IPv4 • *IPv6
		937	Previous IP address [4]	Char(46)	The IP address of the previous console device for *LAN.
		983	Previous device ID [4]	Char(10)	The service tools device ID of the previous console device for *LAN.
		993	Current IP address [4]	Char(46)	The IP address of the current console device for *LAN.
		1039	Current device ID [4]	Char(10)	The service tools device ID of the current console device for *LAN.
		1049	Watch session[5]	Char(10)	Watch session ID.
		1059	Entry[9]	Char(10)	Name of the entry in the entry point table that was changed.
		1069	Related Object[10]	Char(10)	Name of related object. • For Service Tool value LC, this field contains the name of the base entry point table. • For Service Tool value LG, this field contains the name of the replacement program. • For Service Tool value LH, this field contains the name of the compare entry point table.
		1079	Related Object Library[10]	Char(10)	Name of related object library. • For Service Tool value LC, this field contains the name of the base entry point table library. • For Service Tool value LG, this field contains the name of the replacement program library. • For Service Tool value LH, this field contains the name of the compare entry point table library.

Table 225. ST (Service Tools Action) journal entries. QASYSTJE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		1089	Service Tool User ID[11]	Char(10)	Service tools user ID if storage was altered from DST or *DEBUG if storage was altered by a remote service tool debugger.
		1099	User profile[11]	Char(10)	User profile name if storage was altered from SST.
		1109	Address of altered storage[11]	Char(16)	Address of storage that was altered. This is a character representation of the hex address.
		1125	Segment Type[11]	Char(4)	Type of segment that was altered. This is a character representation of the hex value.
		1129	Length of altered storage[11]	Bin(5)	Length of storage that was altered.
		1133	Altered storage[11]	Char(32)	Altered storage value. This is a character representation of the hex value.
		1165	Original storage[11]	Char(32)	Original storage value. This is a character representation of the hex value.

1 This field is only used when the Service Tool value (offset 611) is CE.

2 This field is only used when the Service Tool value (offset 611) is AR or TA.

3 This field is only used when the Service Tool value (offset 611) is ST or OP.

4 This field is only used when the Service Tool value (offset 611) is OP.

5 This field is only used when the Service Tool value (offset 611) is WS or WE.

6 This field is only used when the Service Tool value (offset 611) is AR.

7 This field is only used when the Service Tool value (offset 611) is TA.

8 The Folder Path will contain the 30 character Advanced Analysis Command name when the Service Tool value (offset 611) is GS.

9 This field is only used when the Service Tool value (offset 611) is LG.

10 This field is only used when the Service Tool value (offset 611) is LC, LG, or LH.

11 This field is only used when the Service Tool value (offset 611) is AS.

SV (Action to System Value) journal entries

This table provides the format of the SV (Action to System Value) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_SV table function: AUDIT_JOURNAL_SV

Table 226. SV (Action to System Value) journal entries. QASYSVJE/J4/J5 Field Description File

Offset			Field	Format	Description
JE	J4	J5			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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 **D** Adjustment to Coordinated Universal Time (UTC) **E** Change to option **F** Change to system-wide journal attribute
157	225	611	System Value or Service Attribute	Char(10)	**JRNRCYCNT** Changed journal recovery count value **CACHEWAIT** Changed journal maximum cache wait time **QINPIDCO** Change the current install disk configuration option with QINPIDCO API.
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 before it was changed.

Table 226. SV (Action to System Value) journal entries. QASYSVJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		1621	New Value Continued Extension	Char(1000)	Second continuation of the value to which the system value or service attribute was changed.
		2621	Old Value Continued Extension	Char(1000)	Second continuation of the value of the system value or service attribute before it was changed.

VA (Change of Access Control List) journal entries

This table provides the format of the VA (Change of Access Control List) journal entries. These journal entries are no longer being written to the audit journal.

Table 227. VA (Change of Access Control List) journal entries. QASYVAJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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 on which the event was logged on the network server.
173	241	627	Server Time	Zoned(6,0)	The time when 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.

Table 227. VA (Change of Access Control List) journal entries. QASYVAJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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.

VC (Connection Start and End) journal entries

This table provides the format of the VC (Connection Start and End) journal entries. These journal entries are no longer being written to the audit journal.

Table 228. VC (Connection Start and End) journal entries. QASYVCJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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 on which the event was logged on the network server.
173	241	627	Server Time	Zoned(6,0)	The time when 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.

Table 228. VC (Connection Start and End) journal entries. QASYVCJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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 why 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.

VF (Close of Server Files) journal entries

This table provides the format of the VF (Close of Server Files) journal entries. These journal entries are no longer being written to the audit journal.

Table 229. VF (Close of Server Files) journal entries. QASYVFJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See “Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)” on page 630,“Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)” on page 632, and “Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)” on page 633 for field listing.
156	224	610	Close Reason	Char(1)	The reason why 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.

Table 229. VF (Close of Server Files) journal entries. QASYVFJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
167	235	621	Server Date	Char(6)	The date on which the event was logged on the network server.
173	241	627	Server Time	Zoned(6,0)	The time when the event was logged on the network server.
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.

VL (Account Limit Exceeded) journal entries

This table provides the format of the VL (Account Limit Exceeded) journal entries. These journal entries are no longer being written to the audit journal.

Table 230. VL (Account Limit Exceeded) journal entries. QASYVLJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 for field listing.
156	224	610	Reason	Char(1)	The reason why 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.

Table 230. VL (Account Limit Exceeded) journal entries. QASYVLJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
167	235	621	Server Date	Char(6)	The date on which the event was logged on the network server.
173	241	627	Server Time	Zoned(6,0)	The time when 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.

VN (Network Log On and Off) journal entries

This table provides the format of the VN (Network Log On and Off) journal entries. These journal entries are no longer being written to the audit journal.

Table 231. VN (Network Log On and Off) journal entries. QASYVNJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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
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 on which the event was logged on the network server.
173	241	627	Server Time	Zoned(6,0)	The time when 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.

Table 231. VN (Network Log On and Off) journal entries. QASYVNJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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 why 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

VO (Validation List) journal entries

This table provides the format of the VO (Validation List) journal entries.

Table 232. VO (Validation List) journal entries. QASYVOJ4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	1	1			Heading fields common to all entry types. See “Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)” on page 630 and“Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)” on page 632 for field listing.

Table 232. VO (Validation List) journal entries. QASYVOJ4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
	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
	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 that 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.

Table 232. VO (Validation List) journal entries. QASYVOJ4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	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

VP (Network Password Error) journal entries

This table provides the format of the VP (Network Password Error) journal entries.

Table 233. VP (Network Password Error) journal entries. QASYVPJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 for field listing.

Table 233. VP (Network Password Error) journal entries. QASYVPJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
156	224	610	Error Type	Char(1)	The type of error that occurred. **P** Password error **D** NetServer user disabled
157	225	611	Server Name	Char(10)	The name of the network server description that registered the event. ***NETSERVER**
167	235	621	Server Date	Char(6)	The date on which the event was logged on the network server.
173	241	627	Server Time	Zoned(6,0)	The time when the event was logged on the network server.
179	247	633	Computer Name	Char(8)	The name of the computer initiating the request. This field is no longer used and will contain blanks.
187	255	641	User	Char(10)	The name of the user.
		651	Long Computer Name	Char(46)	The name or IP address of the computer initiating the request.

VR (Network Resource Access) journal entries

This table provides the format of the VR (Network Resource Access) journal entries. These journal entries are no longer being written to the audit journal.

Table 234. VR (Network Resource Access) journal entries. QASYVRJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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.

Table 234. VR (Network Resource Access) journal entries. QASYVRJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
167	235	621	Server Date	Char(6)	The date on which the event was logged on the network server.
173	241	627	Server Time	Zoned(6,0)	The time when 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.
211	279	665	Resource Name	Char(260)	Name of the resource being used.

VS (Server Session) journal entries

This table provides the format of the VS (Server Session) journal entries. These journal entries are no longer being written to the audit journal.

Table 235. VS (Server Session) journal entries. QASYVSJE/J4/J5 field Description File

Offset			Field	Format	Description
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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

Table 235. VS (Server Session) journal entries. QASYVSJE/J4/J5 field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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

VU (Network Profile Change) journal entries

This table provides the format of the VU (Network Profile Change) journal entries. These journal entries are no longer being written to the audit journal.

Table 236. VU (Network Profile Change) journal entries. QASYVUJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See “Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)” on page 630,“Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)” on page 632, and “Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)” on page 633 for field listing.
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 on which the event was logged on the network server.
173	241	627	Server Time	Zoned(6,0)	The time when the event was logged on the network server.

Table 236. VU (Network Profile Change) journal entries. QASYVUJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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.

VV (Service Status Change) journal entries

This table provides the format of the VV (Service Status Change) journal entries. These journal entries are no longer being written to the audit journal.

Table 237. VV (Service Status Change) journal entries. QASYVVJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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

Table 237. VV (Service Status Change) journal entries. QASYVVJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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 on which the event was logged on the network server.
173	241	627	Server Time	Zoned(6,0)	The time when 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.
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.

X0 (Network Authentication) journal entries

This table provides the format of the X0 (Network Authentication) journal entries.

Table 238. X0 (Network Authentication) journal entries. QASYX0JE/J4/J5 Field Description File

Offset			Field	Format	Description
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630, "Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 for field listing.

Table 238. X0 (Network Authentication) journal entries. QASYX0JE/J4/J5 Field Description File *(continued)*					
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

Table 238. X0 (Network Authentication) journal entries. QASYX0JE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
					(continued) **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
	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

Table 238. X0 (Network Authentication) journal entries. QASYX0JE/J4/J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
	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

Table 238. X0 (Network Authentication) journal entries. QASYX0JE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	1641	2027	Client Principal	Char(512)	Client principal from authenticator
	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

Table 238. X0 (Network Authentication) journal entries. QASYX0JE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	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

X1 (Identity Token) journal entries

This table provides the format of the X1 (Identity Token) journal entries.

Table 239. X1 (Identity Token) journal entries. QASYX1J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630, "Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 for field listing.

Table 239. X1 (Identity Token) journal entries. QASYX1J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		610	Entry Type	Char(1)	The type of entry: **D** Delegate of identity token was successful **F** Delegate of identity token failed **G** Get user from identity token was successful **U** Get user from identity token failed
		611	Reason Code	Binary (5)	Reason code for failed request: **9** Token length mismatch **10** EIM identifier mismatch **11** Application instance ID mismatch **12** Token signature not valid **13** Identity token not valid **14** Target user not found **16** Key handle not valid **17** Token version not supported **18** Public key not found **Note:** On a failure, only the information that has been validated up to the point of failure will be filled in the text fields.
		615	Reserved	Char(7)	Reserved
		622	Data CCSID	Binary(5)	The CCSID of the data in the text fields
		626	Receiver length	Binary(5)	The length of the data in the receiver field.
		630	Receiver	Char(510)[1]	The receiver of the identity token that either failed the request or was successful. The data in this field will be in the format: <EIMID>receiver_eimID </EIMID> <APPID>RECEIVER_appID </APPID> <TIMESTAMP>receiver_timestamp </TIMESTAMP>. The timestamp will only be included on delegate requests.

Table 239. X1 (Identity Token) journal entries. QASYX1J5 Field Description File *(continued)*

Offset			Field	Format	Description
JE	**J4**	**J5**			
		1140	Sender Length	Binary(5)	The length of the data in the sender field.
		1144	Sender	Char(510)[1]	The last sender of the identity token that either failed the request or was successful. The data in this field will be in the format The data in this field will be in the format: <EIMID>sender_eimID</EIMID> <APPID>sender_appID</APPID> <TIMESTAMP>sender_timestamp</TIMESTAMP>
		1654	Initiator Length	Binary(5)	The length of the data in the initiator field.
		1658	Initiator	Char(510)[1]	The initiator of the identity token request. If the sender and initiator are the same, the initiator length field will be 0. The data in this field will be in the format: <EIMID>initiator_eimID</EIMID> <APPID>initiator_appID</APPID> <TIMESTAMP>initiator_timestamp</TIMESTAMP>
		2168	Chain Length	Binary(5)	The length of the data in the chain field.
		2172	Chain	Char(2038)[1]	The chain of senders between the initiator and the last sender. The chain will be in the order of latest to earliest. If there are no other senders, then the chain length field will be 0. This field will be truncated if the chain is longer than the length of this field. The data in this field will be in the format: <SNDRz><EIMID>sndrz_eimID</EIMID> <APPID>sndrz_appID</APPID> <TIMESTAMP>sndrz_timestamp </TIMESTAMP> </SNDRz> <SNDRy>...</SNDRy>...
		4210	Chain Entries	Binary(5)	The number of entries in the chain field.
		4214	Chain Entries Available	Binary(5)	The number of available entries for the chain of senders. This number might be greater than the number of entries in the field if the chain field is truncated.
		4218	Source Registry Length	Binary(5)	The length of the data in the source registry field.
		4222	Source Registry	Char(510)[1]	The source registry specified in the identity token.
		4732	Source Registry User Length	Binary(5)	The length of the data in the source registry user field.
		4736	Source Registry User	Char(510)[1]	The source registry user specified in the identity token.

Table 239. X1 (Identity Token) journal entries. QASYX1J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		5246	Target Registry Length	Binary(5)	The length of the data in the target registry field.
		5250	Target Registry	Char(510)[1]	The target registry specified.
		5760	Target Registry User Length	Binary(5)	The length of the data in the target registry user field.
		5764	Target Registry User	Char(510)[1]	The target registry user to which the identity token maps.

1
This is a variable length field. The first 2 bytes contain the length of the field.

X2 (Query Manager Profile Changes) journal entries

The X2 (Query Manager Profile Changes) journal entries do not have a model database outfile.

For information on X2 journal entries see IBM Support, Query Manager Profile Auditing.

XD (Directory Server Extension) journal entries

This table provides the format of the XD (Directory Server Extension) journal entries.

Table 240. XD (Directory Server Extension) journal entries. QASYXDJ5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
		1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630, "Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 for field listing.
		610	Entry Type	Char(1)	The type of entry: **G** Group names. Field 1 through Field 5 contain group names.
		611	Cross Reference	Char(36)	Cross reference string used to correlate this entry with the DI entry using these groups. More than one DI entry can refer to this XD entry if multiple LDAP requests use the same set of groups.
		647	Reserved	Char(100)	

Table 240. XD (Directory Server Extension) journal entries. QASYXDJ5 Field Description File *(continued)*					
Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
		747	Field 1 CCSID	Bin(5)	The CCSID value for field 1.
		751	Field 1 Length	Bin(4)	The length of the data in field 1.
		753	Field 1	Char(2002)	Field 1 data For entry type G, this field will contain a group name from a group membership assertion.
		2755	Field 2 CCSID	Bin(5)	The CCSID value for field 2.
		2759	Field 2 Length	Bin(4)	The length of the data in field 2.
		2761	Field 2	Char(2002)	Field 2 data For entry type G, this field will contain a group name from a group membership assertion.
		4763	Field 3 CCSID	Bin(5)	The CCSID value for field 3.
		4767	Field 3 Length	Bin(4)	The length of the data in field 3.
		4769	Field 3	Char(2002)	Field 3 data For entry type G, this field will contain a group name from a group membership assertion.
		6771	Field 4 CCSID	Bin(5)	The CCSID value for field 4.
		6775	Field 4 Length	Bin(4)	The length of the data in field 4.
		6777	Field 4	Char(2002)	Field 4 data For entry type G, this field will contain a group name from a group membership assertion.
		8779	Field 5 CCSID	Bin(5)	The CCSID value for field 5.
		8783	Field 5 Length	Bin(4)	The length of the data in field 5.
		8785	Field 5	Char(2002)	Field 5 data For entry type G, this field will contain a group name from a group membership assertion.

YC (Change to DLO Object) journal entries

This table provides the format of the YC (Change to DLO Object) journal entries.

Table 241. YC (Change to DLO Object) journal entries. QASYYCJE/J4/J5 Field Description File

Offset			Field	Format	Description
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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 "Numeric codes for access types" on page 890 for a list of the codes for access types.

YR (Read of DLO Object) journal entries

This table provides the format of the YR (Read of DLO Object) journal entries.

Table 242. YR (Read of DLO Object) journal entries. QASYYRJE/J4/J5 Field Description File

Offstes			Field	Format	Description
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630, "Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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 "Numeric codes for access types" on page 890 for a list of the codes for access types.

ZC (Change to Object) journal entries

This table provides the format of the ZC (Change to Object) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_ZC table function: AUDIT_JOURNAL_ZC

Table 243. ZC (Change to Object) journal entries. QASYZCJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630,"Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 for field listing.
156	224	610	Entry Type	Char(1)	Object access **C** Change of an object **U** Upgrade of open access to 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]

Table 243. ZC (Change to Object) journal entries. QASYZCJE/J4/J5 Field Description File *(continued)*					
Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
188	256	642	Access Specific Data	Char(50)	Specific data about the access When the object type is *IMGCLG, this field contains the following format: **Char 3** Index number of the image catalog entry. **Blank** Indicates the operation was against an image catalog. **Char 32** Volume ID of the image catalog entry. **Blank** Indicates the operation was against an image catalog. **Char 1** Access type for the entry. The possible values are listed below. **Blank** Indicates the operation was against an image catalog. **R** The file containing the image catalog entry is read-only. **W** The file containing the image catalog entry is read/write capable. **Char 1** The write protection for the entry. **Blank** Indicates the operation was against an image catalog. **Y** The file containing the image catalog entry is write protected. **N** The file containing the image catalog entry is not write protected.

Table 243. ZC (Change to Object) journal entries. QASYZCJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
					(continued) **Char 10** The name of the virtual device. **Blank** Indicates the operation was against an image catalog or the image catalog is not in Ready status. **Char 3** Not used. When the object type is an integrated file system object, this field contains further information identifying the change request. See the QSYSINC include file, QP0LJRNL.H for the possible values.
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 path name.
	917	1303	Path Name Country or Region ID	Char(2)	The Country or Region ID for the path name.

Table 243. ZC (Change to Object) journal entries. QASYZCJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
	919	1305	Path Name Language ID	Char(3)	The language ID for the path name.
	922	1308	Path Name Length	Binary(4)	The length of the path name.
	924	1310	Path Name Indicator	Char(1)	Path name indicator: **Y** The Path Name field contains complete absolute path name for the object. **N** The Path Name field does not contain an absolute path name for the object, instead it contains a relative path name. The Relative Directory File ID field is valid and can be used to form an absolute path name with this relative path name.
	925	1311	Relative Directory File ID[4]	Char(16)	When the Path Name Indicator field is N, this field contains the file ID of the directory that contains the object identified in the Path Name field. Otherwise it contains hex zeros.[4]
	941	1327	Path Name[5]	Char(5002)	The path name of the object.

1
See “Numeric codes for access types” on page 890 for a list of the codes for access types.

2
These fields are used only for objects in the "root" (/), QOpenSys, 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
If the Path Name Indicator field is N, but the Relative Directory File ID is hex zeros, then there was some error in determining the path name information.

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.

ZR (Read of Object) journal entries

This table provides the format of the ZR (Read of Object) journal entries.

Information from this audit journal entry can be queried with the SYSTOOLS.AUDIT_JOURNAL_ZR table function: AUDIT_JOURNAL_ZR

Table 244. ZR (Read of Object) journal entries. QASYZRJE/J4/J5 Field Description File					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
1	1	1			Heading fields common to all entry types. See "Standard heading fields for audit journal entries QJORDJE5 Record Format (*TYPE5)" on page 630, "Standard heading fields for audit journal entries QJORDJE4 Record Format (*TYPE4)" on page 632, and "Standard heading fields for audit journal entries QJORDJE2 Record Format (*TYPE2)" on page 633 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]

Table 244. ZR (Read of Object) journal entries. QASYZRJE/J4/J5 Field Description File *(continued)*					
Offset					
JE	**J4**	**J5**	**Field**	**Format**	**Description**
188	256	642	Access Specific Data	Char(50)	Specific data about the access. When the object type is *IMGCLG, this field contains the following format: **Char 3** Index number of the image catalog entry. **Blank** Indicates the operation was against an image catalog. **Char 32** Volume ID of the image catalog entry. **Blank** Indicates the operation was against an image catalog. **Char 1** Access type for the entry. The possible values are listed below. **Blank** Indicates the operation was against an image catalog. **R** The file containing the image catalog entry is read-only. **W** The file containing the image catalog entry is read/write capable. **Char 1** The write protection for the entry. **Blank** Indicates the operation was against an image catalog. **Y** The file containing the image catalog entry is write protected. **N** The file containing the image catalog entry is not write protected. **Char 10** The name of the virtual device. **Blank** Indicates the operation was against an image catalog or the image catalog is not in Ready status. **Char 3** Not used.

Table 244. ZR (Read of Object) journal entries. QASYZRJE/J4/J5 Field Description File *(continued)*					
Offset			**Field**	**Format**	**Description**
JE	**J4**	**J5**			
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 path name.
	917	1303	Path Name Country or Region ID	Char(2)	The Country or Region ID for the path name.
	919	1305	Path Name Language ID	Char(3)	The language ID for the path name.
	922	1308	Path Name Length	Binary(4)	The length of the path name.

Table 244. ZR (Read of Object) journal entries. QASYZRJE/J4/J5 Field Description File (continued)					
Offset					
JE	J4	J5	Field	Format	Description
	924	1310	Path Name Indicator	Char(1)	Path name indicator: **Y** The Path Name field contains complete absolute path name for the object. **N** The Path Name field does not contain an absolute path name for the object, instead it contains a relative path name. The Relative Directory File ID field is valid and can be used to form an absolute path name with this relative path name.
	925	1311	Relative Directory File ID[4]	Char(16)	When the Path Name Indicator field is N, this field contains the file ID of the directory that contains the object identified in the Path Name field. Otherwise it contains hex zeros.[4]
	941	1327	Path Name[5]	Char(5002)	The path name of the object.

1
See "Numeric codes for access types" on page 890 for a list of the codes for access types.

2
These fields are used only for objects in the "root" (/), QOpenSys, 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
If the Path Name Indicator field is N, but the Relative Directory File ID is hex zeros, then there was some error in determining the path name information.

5
This is a variable length field. The first 2 bytes contain the length of the path name.

Numeric codes for access types

This table lists the access codes used for object auditing journal entries in files QASYYCJE/J4/J5, QASYYRJE/J4/J5, QASYZCJE/J4/J5, and QASYZRJE/J4/J5.

Table 245. Numeric codes for access types					
Code	**Access type**	**Code**	**Access type**	**Code**	**Access type**
1	Add	26	Load	51	Send
2	Activate Program	27	List	52	Start
3	Analyze	28	Move	53	Transfer
4	Apply	29	Merge	54	Trace
5	Call or TFRCTL	30	Open	55	Verify
6	Configure	31	Print	56	Vary
7	Change	32	Query	57	Work

Table 245. Numeric codes for access types (continued)

Code	Access type	Code	Access type	Code	Access type
8	Check	33	Reclaim	58	Read/Change DLO Attribute
9	Close	34	Receive	59	Read/Change DLO Security
10	Clear	35	Read	60	Read/Change DLO Content
11	Compare	36	Reorganize	61	Read/Change DLO all parts
12	Cancel	37	Release	62	Add Constraint
13	Copy	38	Remove	63	Change Constraint
14	Create	39	Rename	64	Remove Constraint
15	Convert	40	Replace	65	Start Procedure
16	Debug	41	Resume	66	Get Access on **OOPOOL
17	Delete	42	Restore	67	Sign object
18	Dump	43	Retrieve	68	Remove all signatures
19	Display	44	Run	69	Clear a signed object
20	Edit	45	Revoke	70	MOUNT
21	End	46	Save	71	Unload
22	File	47	Save with Storage Free	72	End Rollback
23	Grant	48	Save and Delete		
24	Hold	49	Submit		
25	Initialize	50	Set		

Appendix G. Commands and menus for security commands

The SECTOOLS (Security Tools) menu, the SECBATCH (Submit or Schedule Security Reports to Batch) menu, the Configure System Security (CFGSYSSEC) and Revoke Public Authority (RVKPUBAUT) commands are four security tools you can use to configure your system security.

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

You can use the Security Tools (SECTOOLS) menu to simplify the management and control of the security on your system with plenty of options and commands that it provides.

This figure shows 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
     9. Print profile internals
```

Table 246 on page 893 describes these menu options and the associated commands:

Table 246. 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 246. 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 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 or CHGUSRPRF command to set up user profiles to expire.	
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). Use the DSPEXPSCD command to display the user profiles that are scheduled to expire.	
9	PRTPRFINT	Use the Print Profile Internals command to print a report of internal information about the number of entries in a user profile (*USRPRF) object.	

Table 246. Tool commands for user profiles (continued)			
Menu[1] option	**Command name**	**Description**	**Database file used**
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 247 on page 895 describes the menu options and associated commands for security auditing:

Table 247. 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) and QAUDLVL2 (audit level extension) system values. 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, make 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.	
12	CPYAUDJRNE	Use the Copy Audit Journal Entries command to copy entries from the security audit journal to an output file.	QASYxxJ5[2]
1 Options are from the SECTOOLS menu. **2** xx is the two-character journal entry type. For example, the model output file for AE journal entries is QSYS/QASYAEJ5. The model output files are described in Appendix F, “Layout of audit journal entries,” on page 629 of this topic collection.			

Related concepts

Using CHGSECAUD to set up security auditing

How to use the Security Batch menu

You can use the security batch menu to submit one or more of the Security Tools reports to a job queue to be run later as a batch job. You can also choose to schedule any of the Security Tools reports as batch jobs to be submitted once or to be submitted at regular intervals. Examples in this topic demonstrate how to use the security batch menu.

Here 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 example:

```
                              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

This table describes the menu options and the 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 248. 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]

Table 248. Commands for security reports (continued)			
Menu[1] option	**Command name**	**Description**	**Database file used**
2, 41	DSPAUDJRNE[6]	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.	QASYxxJ5[3]
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 for 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]

Table 248. Commands for security reports (continued)

Menu[1] option	Command name	Description	Database file used
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]
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]

Table 248. Commands for security reports (continued)			
Menu[1] option	**Command name**	**Description**	**Database file used**
24, 63	PRTQAUT	Use the Print Queue Authority command 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]
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]

Table 248. Commands for security reports (continued)			
Menu[1] option	**Command name**	**Description**	**Database file used**
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 about the number of entries contained in a user profile (*USRPRF) object.	
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.	

Table 248. Commands for security reports (continued)			
Menu[1] option	**Command name**	**Description**	**Database file used**

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/QASYAEJ5. The model output files are described in Appendix F, "Layout of audit journal entries," on page 629 of this topic collection.

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.

6

The DSPAUDJRNE command cannot process all security audit record types, and the command does not list all the fields for the records it does support.

Commands for customizing security

This table describes the commands that you can use to customize the security on your system, which are on the SECTOOLS menu.

Table 249. 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" on page 903 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 906 lists the actions that the RVKPUBAUT command performs.	

1

Options are from the SECTOOLS menu.

Related information

Complete the security wizard

Values that are set by the Configure System Security command

This table lists the system values that are set when you run the Configure System Security (**CFGSYSSEC**) command that runs a program that is called QSYS/QSECCFGS.

Table 250. Values set by the CFGSYSSEC command

System value name	Setting	System value description
QALWJOBITP	0 (Do not allow)	Allow jobs to be interrupted
QALWOBJRST	*NONE	Whether system state programs and programs that adopt authority can be restored
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.
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
QFRCCVNRST	4 (Convert objects with sufficient creation data and not valid signatures)	Force conversion on restore
QINACTITV	60	Time period before the system takes action on an 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.
QPWDCHGBLK	3	Number of hours to block a password change
QPWDEXPITV	60	How often users must change their passwords
QPWDEXPWRN	14	Number of days prior to password expiration to begin showing warning
QPWDMINLEN	6 (See note 3 and 5)	Minimum length for passwords
QPWDMAXLEN	8 (See note 4 and 5)	Maximum length for passwords
QPWDPOSDIF	1 (Yes) (See note 5)	Whether every position in a new password must differ from the same position in the last password

Table 250. Values set by the CFGSYSSEC command (continued)		
System value name	**Setting**	**System value description**
QPWDLMTCHR	See note 2 and 5	Characters that are not allowed in passwords
QPWDLMTAJC	1 (Yes) (See note 5)	Whether adjacent numbers are prohibited in passwords
QPWDLMTREP	2 (Cannot be repeated consecutively) (See note 5)	Whether repeating characters in are prohibited in passwords
QPWDRQDDGT	1 (Yes) (See note 5)	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
QPWDRULES	• *MINLEN6 • *MAXLEN10 • *LMTSAMPOS • *LMTPRFNAME • *DGTMIN1 • *CHRLMTAJC • *DGTLMTAJC • *DGTLMTFST • *DGTLMTLST • *SPCCHRLMTAJC • *SPCCHRLMTFST • *SPCCHRLMTLST (see note 6)	Rules for forming a valid password.
QPWDVLDPGM	*NONE	The user exit program that the system calls to validate passwords
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	50	The level of security that is enforced
QVFYOBJRST	3	Verify object on restore

Table 250. Values set by the CFGSYSSEC command (continued)

System value name	Setting	System value description

Notes:

1. If you are currently running with a QSECURITY value of 30 or lower, be sure to review the information in Chapter 2, "Using System Security (QSecurity) system value," on page 7 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.
3. If the minimum length for passwords is already greater than 6, the QPWDMINLEN system value will not be changed.
4. If the maximum length for passwords is already greater than 8, the QPWDMAXLEN system value will not be changed.
5. This system value is only changed when the QPWDRULES system value currently specifies a value of *PWDSYSVAL.
6. This system value will not be changed if its current value is *PWDSYSVAL.

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 system values of the settings are not appropriate for your installation, you can create your own version of the program that processes the Configure System Security (**CFGSYSSEC**) command.

To change the program, perform the following steps:

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.
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.
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 need to type the following command:

   ```
   CHGCMD CMD(QSYS/CFGSYSSEC) PGM(QGPL/MYSECCFG)
   ```

   Notes:

   a. 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.

b. If you change the **RVKPUBAUT** command to use a different command processing program, then the digital signature of this command will no longer be valid.

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 251 on page 906 and the application programming interfaces (APIs) that are listed in Table 252 on page 906. When your system arrives, these commands and APIs have their public authority set to *USE.

The commands that are listed in Table 251 on page 906 and the APIs that are listed in Table 252 on page 906 all perform functions on your system that might 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 251. Commands whose public authority is set by the RVKPUBAUT command

Command		
ADDAJE ADDCFGLE ADDCMNE ADDJOBQE ADDPJE ADDRTGE ADDWSE CHGAJE CHGCFGL CHGCFGLE CHGCMNE CHGCTLAPPC CHGDEVAPPC	CHGJOBQE CHGPJE CHGRTGE CHGSBSD CHGWSE CPYCFGL CRTCFGL CRTCTLAPPC CRTDEVAPPC CRTSBSD ENDRMTSPT RMVAJE RMVCFGLE	RMVCMNE RMVJOBQE RMVPJE RMVRTGE RMVWSE RSTLIB RSTOBJ RSTS36F RSTS36FLR RSTS36LIBM STRRMTSPT STRSBS WRKCFGL

The APIs in Table 252 on page 906 are all in the QSYS library:

Table 252. Programs whose public authority is set by the RVKPUBAUT command

API		
QTIENDSUP QTISTRSUP QWTCTLTR QWTSETTR QY2FTML		

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 the settings are not appropriate for your installation, you can create your own version of the program that processes the Revoke Public Authority (**RVKPUBAUT**) command.

To change the program, perform the following steps:

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*.
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.
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 need to type the following command:

   ```
   CHGCMD CMD(QSYS/RVKPUBAUT) PGM(QGPL/MYRVKPGM)
   ```

   Notes:

   a. 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.

   b. If you change the **RVJPUDAUT** command to use a different command processing program, then the digital signature of this command will no longer be valid.

Appendix H. Related information for IBM i security reference

Listed here are the product manuals and IBM Redbooks® (in PDF format), Web sites, and information center topics that relate to the security topic. You can view or print any of the PDFs.

Manuals

- Recovering your system, provides information about planning a backup and recovery strategy, saving information from your system, and recovering your system, auxiliary storage pools, and disk protection options.
- Installing, upgrading, or deleting IBM i and related softwareprovides step-by-step procedures for initial install, installing licensed programs, program temporary fixes (PTFs), and secondary languages from IBM.
- Remote Workstation Support, provides information about how to set up and use remote workstation support, such as display station pass-through, distributed host command facility, and 3270 remote attachment.
- Cryptographic Support/400, describes the data security capabilities of the Cryptographic Facility licensed program. It explains how to use the facility and provides reference information for programmers.
- Local Device Configuration, provides information about how to do an initial configuration and how to change that configuration. It also contains conceptual information about device configuration.
- *SNA Distribution Services*, SC41-5410 (2,259 KB), 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. (This manual is not included in this release of the IBM i Information Center. However, it might be a useful reference to you. The manual is available from the IBM Publications Center as a printed hardcopy that you can order or in an online format that you can download at no charge.)
- *ADTS for AS/400: Source Entry Utility*, SC09-2605, 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. (This manual is not included in this release of the IBM i Information Center. However, it might be a useful reference to you. The manual is available from the IBM Publications Center as a printed hardcopy that you can order or in an online format that you can download at no charge.)

Web sites

- Lotus Documentation (http://www.lotus.com/ldd/doc)

  This Web site provides information about Lotus Notes, Domino®, and IBM Domino for IBM i. 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.

Other information

- Planning and setting up system security 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 tools that are part of IBM i.
- *Implementing AS/400 Security, 4th Edition* (October 15, 2000) by Wayne Madden and Carol Woodbury. Loveland, Colorado: 29th Street Press. Provides guidance and practical suggestions for planning, setting up, and managing your system security.

  ISBN Order Number
  1583040730
- IBM i Access for Windows provides technical information about the IBM i Access for Windows programs for all versions of IBM i Access for Windows
- TCP/IP setup provides information that describes how to use and configure TCP/IP.
- TCP/IP applications, protocols, and services provides information that describes how to use TCP/IP applications, such as FTP, SMTP, and TELNET.
- Basic system operations provides information about how to start and stop the system and work with system problems.
- Integrated file system provides an overview of the integrated file system, including what it is, how it can be used, and what interfaces are available.
- iSeries and Internet security 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. Optical storage provides information about 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.
- Printing provides information about printing elements and concepts of the system, printer file and print spooling support for printing operation, and printer connectivity.
- Control language 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.

  It also provides a description of all the iSeries control language (CL) and its IBM i commands. The IBM i commands are used to request functions of the IBM i (5722-SS1) licensed program. All the non-IBM i 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.
- Programming provides information about many of the languages and utilities available on the iSeries. It contains summaries of:
  - All iSeries CL commands (in IBM i 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).
- Systems management includes information about performance data collection, system values management, and storage management.

- Database file concepts provides an overview of how to design, write, run, and test the statements of Db2 Query Manger and SQL Development Kit for IBM i. 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. It also provides information about 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

Saving PDF files

To save a PDF on your workstation for viewing or printing:

1. Right-click the PDF in your browser (right-click the link above).
2. Click the option that saves the PDF locally.
3. Navigate to the directory in which you want to save the PDF.
4. Click **Save**.

Downloading Adobe Reader

You need Adobe Reader installed on your system to view or print these PDFs. You can download a free copy from the Adobe Web site (www.adobe.com/products/acrobat/readstep.html).

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 grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country 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.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
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 document 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 measurements 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.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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 illustrate 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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

This Security reference publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be

trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux® is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks of Oracle, Inc. in the United States, other countries, or both.

Other product and service names might be trademarks of IBM or other companies.

Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

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

Product Number: 5770-SS1

SC41-5302-14