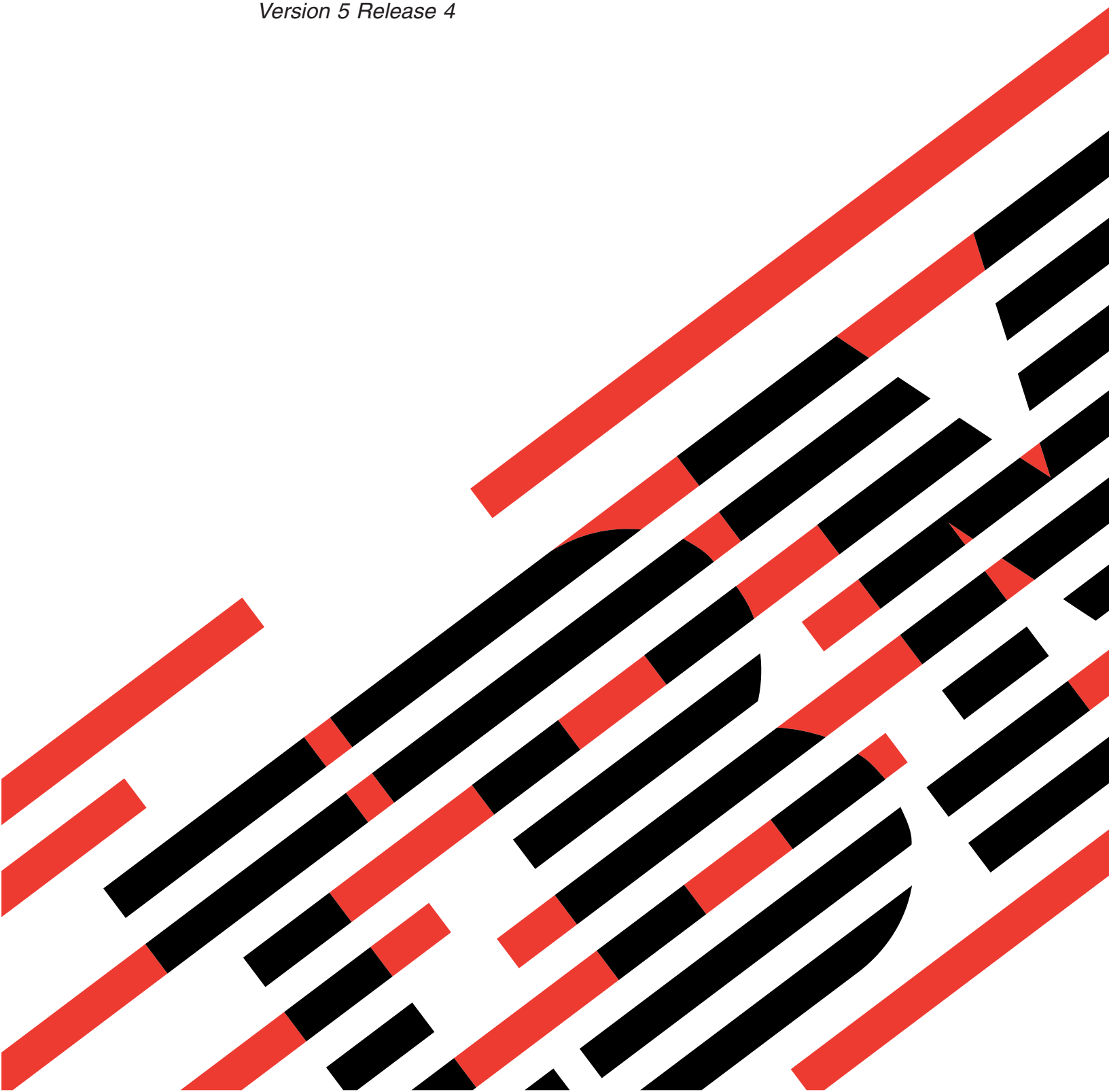
IBM Systems - iSeries

Security
Plan and set up system security

Version 5 Release 4



IBM Systems - iSeries

Security
Plan and set up system security

Version 5 Release 4

Note

Before using this information and the product it supports, be sure to read the information in "Notices," on page 301.

First Edition (February 2006)

This edition applies to version 5, release 4, modification 0 of IBM i5/OS (product number 5722-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.

Contents

Plan and set up system security

This topic collection provides you with detailed information about planning, setting up, and using your system security. This topic collection combines the information formerly in the Basic system security and planning topic collection and in the *Tips and Tools for Securing Your iSeries*™ manual.

Determining your company's system security is one of the most basic and most important decisions that you will make during the course of building your security plan. With system security you need to balance the need to safeguard your valuable information and the need of users to access that information to successfully make your company thrive. To strike this balance you must understand the specific needs and goals of your company's current direction but also be aware of future needs. Your security plan must protect your resources but also must be flexible enough to grow as your company grows.

Several tools exist that can aid you in creating, configuring, and managing your system-level security on your server. It is important to note that security does not end with protecting the server and managing access to assets that are stored on the system. A complete security implementation needs to cover not only system level security, but also network level security and transaction level security. This topic focuses on system-level security.

Use this information to develop a personalized plan that fits your company's specific system security needs. After you complete the planning phase of your system security, you can set up system security by using the instructions provided in this information.

What's new for V5R4

Plan and set up system security provides information on how to effectively and systematically plan and configure system-level security.

This new topic combines information from the Basic system security and planning topic and the *Tips and Tools for securing your system* (SC41-5300) manual. Both of these topics have been removed from the Information Center.

How to see what's new or changed

Revision bars are not used in this topic because it is new.

To find other information about what is new or changed this release, see the Memo to users.

Printable PDF

Use this to view and print a PDF of this information.

To view or download the PDF version of this document, select Plan and set up system security (about 3907 KB).

You can view or download these related manuals:

- iSeries Security Reference (13 682 KB)

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 **Save Target As** if you are using Internet Explorer. Click **Save Link As** if you are using Netscape Communicator.
3. Navigate to the directory in which you would like to save the PDF.
4. Click **Save**.

Downloading Adobe Acrobat Reader

You need Adobe Acrobat Reader to view or print these PDFs. You can download a copy from the Adobe Web site (www.adobe.com/products/acrobat/readstep.html) .

Frequently asked questions

These are common questions about setting up and using system security.

Questions that customers often ask

Administrators and security officers are faced with a wide variety of options and solutions for protecting the systems that they manage. All of these potential solutions can be confusing and daunting; however, good system security involves understanding your basic security needs and the role that security plays within your company. To understand the value of security for your company and its systems, you should know what security means to you at its most basic level.

1. **Why is security important?**

   Answer: The information stored on your system is one of your most important business assets. This sensitive information can be customer accounts, payroll statements, and financial statements. You must balance the need for protecting this information with the need to allow your employees access to complete their job responsibilities. You need to keep three important objectives in mind when determining how to protect your information assets:

   - **Confidentiality:** Good security measures can prevent people from seeing and disclosing confidential information. On your systems, what information do you consider confidential, which only a few select individuals can see and maintain?
   - **Integrity:** To some extent, a well-designed security system can ensure the accuracy of the information on your computer. With the right security, you can prevent unauthorized changes or deletions of data.
   - **Availability:** If someone accidentally or intentionally damages data on your system, you cannot access those resources until you recover them. A good security system can prevent this kind of damage.

     When people think about system security, they usually think about protecting their system from people outside the company, such as business rivals. Actually, protection against curiosity or system accidents by proper users is often the greatest benefit of a well-designed security system. In a system without good security features, a user might unintentionally delete an important file. A well-designed security system helps prevent this type of accident.

2. **Who should be responsible for security on my system?**

   Answer: Different companies take different approaches to security. Sometimes programmers have responsibility for all aspects of security. In other cases, the person who manages the system is also in charge of security. To determine who should be responsible for security on your system or systems, consider the suggested approach of:

   - Your method of planning security depends on whether your company purchases or develops applications. If you develop your own applications, communicate your security needs during the development process. If you purchase applications, understand and work with the application designer. In both cases, the people who design applications should consider security as part of the design.

- Your method of planning resource security depends on whether your company purchases or develops applications. If you develop your own applications, communicate your resource security needs during the development process. If you purchase applications, understand and work with the application designer. In both cases, the people who design applications should consider security as part of the design.

3. **Why should I customize security on my system?**

   Answer: A small system might have three to five users that run a few applications. A large system might have thousands of users on a large communications network running many applications. You have the opportunity to change many things about how the system looks to your users and how it performs.

   When your system first arrives, you probably will not need or want to do very much customizing. IBM® ships your system with initial settings, called defaults, for many options. These defaults are the choices that usually work best for new installations.

   Note: All new systems ship with a default security level of 40. This security level ensures that only users who you have defined can use the system. It also prevents potential integrity or security risks from programs that can circumvent security.

   However, if you do some customizing, you can make your system a simpler and more effective tool for your users. For example, you can make sure that a user always gets the correct menu when signing on. You can make sure that every user's reports go to the right printer. Your users will feel that more confident about the system if you do some initial customizing to make it look and feel like their own system.

Questions customers should ask themselves

1. **Have I clearly defined my company's business requirements?**

   Answer: To plan and set up security on your systems effectively, you must first know what your business requires to function effectively and efficiently. You need to understand how your systems will be used within your company. For example, systems that contain critical applications, such as databases that contain your company accounts, would need higher level of security than systems used for testing products within your company.

2. **What assets do I want to protect?**

   Answer: Your business assets comprise not only the physical systems that you manage, but also the data and information that is stored on them. To minimize theft and tampering, you need to create an inventory of your systems and the information that they store.

   The amount of security you need depends on the type of information stored on that system, the sensitivity of that information, and the consequences to your business if that data is stolen or compromised. Understanding the risks that your systems may face allows you to more effectively manage security on your systems.

3. **Do I have a company policy regarding security?**

   Answer: A security policy defines your company's requirements for protecting your company's resources, responding security-related incidents, and conducting secure business transactions with remote employees, business partners, and public customers. This security policy should entail physical security of your systems, network security issues, such as Internet access for employees, and measures for assessing and monitoring security on your systems. Think of your security policy as your foundation for all your security decisions. Your security policy needs to reflect your core business values, but also be flexible enough to accommodate future business demands.

4. **Do my employees have or need access to the Internet?**

   Answer: Today, most companies see the need to allow employees access to the Internet to conduct research and respond to customers related to daily operations of their businesses. Whenever you connect your systems and users to the Internet, your internal resources are at risk of an attack. To protect your network from these risks that are associated with Internet use, you need to decide which network services will be allowed, how users will connect to the Internet, and how network security will be monitored in your network. Any decisions you make regarding the Internet and its use needs

to be clearly defined and communicated to employees within your security policy. It is important to ensure that all your employees understand and sign a compliance agreement with these policies. Although implementing a network security policy is beyond the scope of this topic, you should include information regarding network security in your overall security policy.

Concepts

To effectively create a security policy and plan security measures for your system, you need to understand the following security concepts, some of which are general concepts and some of which are specific to the hardware type.

A small system might have three to five users and a large system might have several thousand users. Some installations have all their workstations in a single, relatively secure area. Others have widely distributed users, including users who connect by dialing in and indirect users connected through personal computers or system networks. Security on this system is flexible enough to meet the requirements of this wide range of users and situations. You need to understand the features and options available so that you can adapt them to your own security requirements. This topic provides an overview of the security features on the system.

System security has three important objectives:

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

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

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

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

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

As you plan security on your system and decide how much security you need, consider these questions:
- Is there a company policy or standard that requires a certain level of security?
- Do the company auditors require some level of security?
- How important is your system and the data on it to your business?
- How important is the error protection provided by the security features?
- What are your company security requirements for the future?

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

Basic terminology

This topic provides users with basic security terminology.

Object

An object is a named space on the system that you or an application can manipulate. Everything on the system that you or an application can work with is considered an object. Objects provide a common interface for working with system components. The most common examples of objects are files and programs. Other types of objects include commands, queues, libraries, and folders. Objects on the system are identified by object name, object type, and the library in which the object resides. You can secure each object on the system.

Library

A library is a special type of object that is used to group other objects. Many objects on the system reside in a library. Libraries are essentially containers, or organizational structures for other objects, and you can use them to reference other objects on your system. Libraries might contain many objects, and might be associated with a specific user profile or application. QSYS, which contains all other libraries on the system, is the only library that can contain other libraries. Objects in a library are handled like objects in a subdirectory. A library cannot live inside a directory.

Directory

A directory is a special object that provides another way to group objects on the system. Objects can reside in a directory and a directory can reside in another directory, forming a hierarchical structure. Each file system is a major **subtree** in the integrated file system directory structure. Directories are different from libraries in that the address of each library maps to the QSYS library while directories are not addressable. Names of libraries are restricted to 10 characters while directories can have longer names which might be case sensitive. Directories can have multiple names because the path to the directory is what is named and not the directory itself. You would use different commands and authority requirements when working with directories and libraries.

User profile

Every system user must have a user identity before they can sign on to and use a system. This user identity is a special object called a user profile, which only an administrator with appropriate system authority can create for a user.

Special authority

Special authority determines whether the user is allowed to perform system functions, such as creating user profiles or changing the jobs of other users.

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. Certain system models are equipped with a keylock that prevents unauthorized functions at the system unit.

Application security

Application security deals with the applications you store on your system and how you will protect those applications while simultaneously allowing users access to them.

Resource security

Resource security on the system allows you to define who can use objects and how objects can be used. The ability to access an object is called **authority**. When you set up object authority, you need to be careful to give your users enough authority to do their work without giving them the

ability to browse and change the system. Object authority gives permissions to the user for a specific object and can specify what the user is allowed to do with the object. An object resource can be limited through specific, detailed user authorities such as adding records or changing records. System resources can be used to give the user access to specific system-defined subsets of authorities: *ALL, *CHANGE, *USE, and *EXCLUDE. System values and user profiles control who has access to your system and prevent unauthorized users from signing on. Resource security controls the actions that authorized system users can perform, and the objects that they can access after they have signed on successfully. Resource security supports the main goals of security on your system to protect:

- Confidentiality of information
- Accuracy of information to prevent unauthorized changes
- Availability of information to prevent accidental or deliberate damage

Security policy

A security policy allows you to implement and manage security on an i5/OS™ system. Use the eServer™ Security Planner to help you plan for and implement a basic security policy for your servers.

Related information

Security terminology

Security levels

Security on your system is arranged in a series of levels, with each level offering a greater degree of security and protection of your data than the previous level.

You can choose how much security you want the system to enforce by setting the security level (QSECURITY) system value. i5/OS supports these fully-integrated system security levels:

- **Level 20: Password security**

  At this security level, users that access to the system must have a password and user ID that the system recognizes. The system administrator creates both the user ID and initial password for users. This level of security allows users total authority to do anything they want on the system, which means that all users can access all data, files, objects, and so on, on your system because all users have *ALLJOB special authority.

- **Level 30: Password and resource security**

  At this security level, resource security is enforced on the system. That is, users must have specific authority to use objects because they do not have any authority by default. Users do not have automatic access to everything on the system and the system administrator must define a valid user ID and password for them. User access is limited by the security policies of the business.

- **Level 40: Integrity protection**

  At this security level, resource security and integrity protection are enforced, and the system itself is protected against users. Integrity protection functions, such as the validation of parameters for interfaces to the operating system, help protect your system and the objects on it from tampering by experienced system users. For example, user-written programs cannot directly access the internal control blocks through pointer manipulation. Level 40 is the default security level for every new installation and is the recommended security level for most installations.

- **Level 50: Advanced integrity protection**

  At this security level, advanced integrity protection is added to the resource security and level 40 integrity protection enforcement. Advanced integrity protection includes further restrictions, such as the restriction of message-handling between system state programs and user state programs. Not only is the system protected against user-written programs, but it ensures that users only have access to data on the system, rather than information about the system itself. This offers greater security against anyone attempting to learn about your system. Level 50 is the recommended level of security for most businesses, because it offers the highest level of security currently possible. Also, level 50 is the required level for C2, FIPS-140, and Common Criteria certifications.

Related concepts

"Plan system security" on page 33
System security entails controlling user access and their privileges, maintaining information integrity, monitoring processes and access, auditing system functions, and providing backup and recovery of security related information.

Lockable security system values

You can lock the security-related system values to prevent users and programs from changing those 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.

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.

Related information

Lock function of security-related system values

Global settings

Global settings affect how work enters the system and how the system appears to other users.

These global settings include the following:

- Security system values, which control security on your system, fall into one of four groups:
  - General security system values
  - Other system values with security properties
  - System values that control passwords
  - System values that control auditing

  Think of system values as company policy. System values apply to everyone using the system, unless something more specific, such as a user profile, overrides the system value. System values allow you to customize many characteristics of your system, including system security characteristics. For example, you can define how many signon attempts to allow at a device, whether the system automatically signs off inactive workstations, how long passwords can be used and changed, and other password characteristics.
- Network attributes, which control how your system participates (or chooses not to participate) in a network with other systems.
- Subsystem descriptions, which determine how work enters the system and what environment the work runs in. A number of work management values have security implications.
- Communications configuration affects how work enters your system. You must protect communication to and from your system with the rest of the network.

Related information

Work management

User profiles

Every system user must have a user identity before they can sign on to and use a system. This user identity is called a user profile.

A user identity is a string of characters that uniquely identifies a user to a system. Only an administrator with appropriate system authority can create a user profile for a user.

A user profile controls what the user can do and customizes the way the system appears to the user. A user profile contains the information that i5/OS requires to allow users to sign on to a system, to access their own customized session, including their own message and output queue, and to access functions and objects to which they have been granted authority. Designing user profiles well can help you protect your system and customize it for your users. Every system user must have a user profile and a system administrator must create the user profile for the user.

There are a number of parameters that an administrator can define for a user profile, including a number of security related attributes. Following are descriptions of a few important security attributes 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.

You can include a user profile in group profiles. In this way, all group members share access to specific objects and share ownership of objects. Group profiles can simplify many user administration tasks by allowing you to apply a single change to many users.

For more information on user profiles, see "Chapter 4. User Profiles" in the *iSeries Security Reference*.

Related concepts

"Plan user profiles" on page 104
This topic describes the purpose of user profiles and how to design them.

"Change a user profile" on page 249
This topic describes how to change a user profile, and provides step-by-step instructions.

"Enable a disabled user profile" on page 249
This topic describes how to enable a disabled user profile, explains why it is important, and provides step-by-step instructions.

Group profiles

Group profiles define authority for a group of users.

You can use a group profile to perform the following tasks:

- Define authority for a group of users, rather than giving authority to each user individually.
- Own objects on the system.
- Use a group profile as a pattern when creating individual user profiles by using the copy profile function.

A group profile is a special type of user profile and can own objects on the system. Typically, you create a group profile for a set of users with similar system access and usage needs. For example, you might create a group profile for a set of users who need to use the same applications in the same way.

You can also use a group profile as a pattern when you create individual user profiles, either by using the copy-profile function or by using the security policies menu to edit user authorities in iSeries Navigator.

A group profile servers two purposes on the system:

- **Security tool**: A group profile provides a simple way to organize who can use certain objects on your system (object authorities). You can define object authorities for an entire group rather than for each individual member of the group.
- **Customization tool**: You can use a group profile 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 in the group profile and copy them to the individual user profiles.

You can use group profiles to make it easier to maintain a simple, consistent scheme for both security and customization.

For more information on working with group profiles, see the following sections in the *iSeries Security Reference*:

"Planning Group Profiles", which discusses using group authority

"Group Ownership of Objects", which discusses what objects should be owned by group profiles.

"Primary Group for an Object", which discusses using primary group and primary group authority for an object

"Copying User Profiles", which describes how to copy a group profile to create an individual user profile.

Related concepts

"Plan group profiles" on page 99
This topic describes the purpose of group profiles and how to design them. Use group profiles to define authorities for a group of users, rather than giving authority to each user individually.

"Create a group profile" on page 184
This article describes how to create a group profile. You can use a group profile to define authority for a group of users, rather than giving authority to each user individually.

Authorization lists

Like a group profile, an authorization list allows you to group objects with similar security requirements and associate the group with a list of users and user authorities.

Authorization lists provide an efficient way to manage the authority to similar objects on the system and aid in the recovery of security information.

Providing each user with explicit access to every object they need to work with might create a great deal of duplicated effort, because many users need to access the same group of objects. A much easier way to provide this access is to create authorization lists. Authorization lists consist of a list of users or groups, the type of authority (*USE, *CHANGE, and *EXCLUDE) for each user or group, and a list of objects to which that this list provides access.

For example, you can create an authorization list to contain a list of objects related to an inventory database. A user responsible for ordering new inventory items can be granted authority to see the contents of the database objects. Additionally, a user group in shipping and receiving needs to update this database as parts come in and out of stock. This group can have authority to change the contents of the objects.

An authorization list has these advantages:

- Authorization lists simplify managing authorities. User authority is defined for the authorization list, not for the individual objects on the list. If a new object is secured by the authorization list, the users on the list gain authority to the object.
- One operation can be used to give a user authority to all the objects on the list.

- Authorization lists reduce the number of private authorities on the system. Each user has a private authority to one object, the authorization list. This gives the user authority to all the objects on the list. Reducing the number of private authorities in the system has the following advantages:
  - Reduces the size of user profiles.
  - Improves the performance when saving the system (SAVSYS) or saving the security data (SAVSECDTA).
- Authorization lists provide a good way to secure files. If you use private authorities, each user will have a private authority for each file member. If you use an authorization list, each user will have only one authority. Also, files that are open cannot have authority granted to the file or revoked from the file. If you secure the file with an authorization list, you can change the authorities, even when the file is open.
- Authorization lists provide a way to remember authorities when an object is saved. When an object is saved that is secured by an authorization list, the name of the authorization list is saved with the object. If the object is deleted and restored to the same system, it is automatically linked to the authorization list again. If the object is restored on a different system, the authorization list is not linked, unless ALWOBJDIF(*ALL) is specified on the restore command.

From a security management view, an authorization list is the preferred method to manage objects that have the same security requirements. Even when there are only a few objects that would be secured by the list, there is still an advantage to using an authorization list instead of using private authorities on the object. Because the authorities are in one place (the authorization list), it is easier to change who is authorized to the objects. It is also easier to secure any new objects with the same security level authorities as the existing objects.

If you use authorization lists, 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 additional system resources; therefore, system 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.

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. The following table shows the characteristics of the two methods.

Table 1. Authorization list and group profile comparison

Usage considerations	Authorization List	Group Profile
Can use 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

Table 1. Authorization list and group profile comparison (continued)

Usage considerations	Authorization List	Group Profile
Can secure all object types	No	Yes
Association with object is deleted when object is deleted	Yes	No
Association with object is saved when the object is saved	Yes	No

You can find more detailed information about authorization lists in "Comparison of group profiles and authorization lists" in the *iSeries Security Reference*.

Related concepts

"Plan authorization lists" on page 140
You can group objects with similar security requirements by using an authorization list.

"Create an authorization list" on page 197
This article describes the task, create an authorization list, explains why it is important, and provides step-by-step instructions.

Validation list objects

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

You can use validation list objects to perform the following tasks:

- Securely store user authentication information for applications.
- Provide an authorization mechanism for users who do not have and do not need an i5/OS user profile, such as internet users.

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 implement the concept of an internet user. Validation lists allow ICS to 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 system 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 system resources or authority to signon or run jobs. A system user profile is never created for the internet users.

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 user passwords, which are encrypted, rather than the application performing the verification itself.

For more information on validation list objects, see Chapter 7, "Planning the use of validation list objects" in the *iSeries Security Reference*.

Menu security

Menu security controls which menu functions a user can perform.

This system was originally designed as a follow-on product for S/36 and S/38. Many system installations were, at one time, S/36 or S/38 installations. To control what users could do, security administrators on those earlier systems often used a technique that is referred to as menu security or menu access control.

Menu access control means that when a user signs on, the user sees a menu. The user can perform only those functions that are on the menu. The user cannot get to a command line on the system to perform any functions that are not on the menu. In theory, the security administrator does not have to worry about authority to objects because menus and programs control what users can do.

Note: Menus are not secure if the system allows any network interfaces to access the system. Most of those interfaces do not know anything about menu security.

Related concepts

"Set up menu security" on page 201
This article discusses the user profile parameters for setting up menu security.

User security

From a user's point of view, security affects how they use and complete tasks on the system.

User security includes how users interact with the system to complete their tasks. It is important to consider how a user will view security. For example, setting passwords to expire every five days might frustrate and interfere with a user's ability to complete his or her job. On the other hand, too lax a password policy might cause security problems.

To provide the right security for your system, you need to divide security into specific parts that you can plan, manage, and monitor. From a user's point of view, you can divide your system security into several parts.

User security includes all areas where security affects the users and where users can affect the system. Key components of user security include:

- **Physical access to the system**

  Physical security protects the system unit and all system devices, including backup storage media, such as diskettes, tapes, or CDs from accidental or intentional loss or damage. Most measures you take to ensure the physical security of your system are external to the system. However, the system ships with a keylock or electronic keystick that prevents unauthorized use of functions at the system unit.
- **How users signon**

  Signon security prevents a person who is not identified on the system from signing on. To sign on, an individual must present valid credentials, such as entering a valid combination of user ID and password. You can use both system values and individual user profiles to make sure that your signon security is not violated. For example, you can require that passwords be changed on a regular basis. You can also prevent the use of passwords that are easy to guess.
- **What users are allowed to do**

  An important role of security, and of system customization, is to define what users can do. From a security perspective, this is often a limiting function, such as preventing people from seeing certain information. From a system customizing perspective, this is an empowering function. A properly customized system makes it possible for people to do their jobs well by eliminating unnecessary tasks and information. Some methods for defining what users can do are appropriate for the security officer, while others are the responsibility of programmers. This information focuses primarily on those things that a security officer usually does. Parameters are available in individual user profiles, job descriptions, and classes to control what the user can do on the system. The list below briefly describes the techniques available:
  - Limiting users to a few functions.

    You can limit users to a specific program, menu or set of menus, and a few system commands based on their user profile. Usually, the security officer creates and controls user profiles.
  - Restricting system functions.

    System functions allow you to save and restore information, manage printer output, and set up new system users. Each user profile specifies which of the most common system functions that the user

can perform. You perform system functions by using control language (CL) commands and APIs. Because every command and API is an object, you can use object authorities to control who can use them and complete system functions.

– Determining who can use files and programs.

  Resource security provides the capability to control the use of every object on the system. For any object, you can specify who can use it and how they can use it. For example, you can specify that one user can only look at the information in a file; another user can change data in the file; a third user can change the file or delete the entire file.

– Preventing abuse of system resources.

  The processing power on your system can become just as important to your business as the data that you store on it. The security officer helps to ensure that users do not misuse system resources by running their jobs at a high priority, printing their reports first, or using too much disk storage.

- How your system communicates with other computers.

  Additional security measures may be necessary if your system communicates with other computers or with programmable workstations. If you do not have proper security controls, someone on another computer in your network can start a job or access information on your computer without going through the signon process. You can use both system values and network attributes to control whether you allow remote jobs, remote access of data, or remote PC access on your system. If you allow remote access, you can specify what security to enforce. You can find descriptions for all system values in Chapter 3, “Security System Values,” of the *iSeries Security Reference*.

- How to save your security information.

  You need to regularly back up the information on your system. In addition to saving the data on your system, you need to save security information. If a disaster occurs, you need to be able to recover information about system users, authorization information, and the information itself.

- How to monitor your security plan.

  The system provides several tools for monitoring security effectiveness:

  – Messages are sent to the system operator when certain security violations occur.

  – Various security-related transactions can be recorded in a special audit journal.

  “Monitor security” on page 269 discusses the use of these tools in general terms. You can find more details on security auditing in Chapter 9, ″Auditing Security on the System,″ in the *iSeries Security Reference*.

- How to customize the security on your system.

  You can customize your system to help your users accomplish their daily work. To best customize your system for your users, think of what they need to accomplish their work successfully. You can customize the system to show menus and applications in several ways:

  – Show users what they want to see.

    Most of us arrange our desks and our offices so we can easily reach the things that we need most. Think of your users’ access to the system in the same way. After signing on to the system, a user should first see the menu or display that person uses the most. You can easily design user profiles to make this happen.

  – Eliminate unnecessary applications.

    Most systems have many different applications on them. Most users only want to see the things they need to do their jobs. Limiting them to a few functions on the system makes their jobs easier. With user profiles, job descriptions, and appropriate menus, you can give each user a specific view of the system.

  – Send something to the right output location.

    Users should not have to worry about how to get their reports to the correct printer or how their batch jobs should run. System values, user profiles, and job descriptions do these things.

  – Provide assistance.

No matter how well you succeed in customizing the system, users may still wonder “Where is my report?” or “Has my job run yet?” Operational Assistant displays provide a simple interface to system functions, which help users answer these questions. Different versions of system displays, called assistance levels, provide help for users with different levels of technical experience. When your system arrives, Operational Assistant displays are automatically available for all users. However, the design of your applications may require you to change the way users get access to the Operational Assistant menu. The system provides tools which allow you to customize your system security to protect your resources while allowing users to access those resources.

Related concepts

“Plan user security” on page 97
Planning user security includes planning all areas where security affects the users on your system.

“Set up user security” on page 180
Setting up user security involves installing application libraries, and setting up user groups and profiles.

“Save security information” on page 236
This topic presents an overview of how you save and restore security information.

Related information

Backup and Recovery PDF

Resource security

You can use resource security on the system to control the actions of authorized users after successful authentication.

System values and user profiles control who has access to your system and prevent unauthorized users from signing on. Resource security controls the actions that authorized system users can perform after they have signed on successfully. Resource security supports the main goals of security on your system to protect:

- Confidentiality of information.
- Accuracy of information to prevent unauthorized changes.
- Availability of information to prevent accidental or deliberate damage.

The security officer protects the resources (objects) on the system by determining who has the authority to use them and how user can access these objects. The security officer can set object authorities for individual objects or for groups of objects (authorization lists). Files, programs, and libraries are the most common objects requiring protection, but system security allows you to set object authorities for any object on the system.

You can manage resource security simply and effectively, if you plan a straightforward approach in advance. A resource security scheme created without prior planning can become complicated and ineffective.

Resource security on the system allows you to define who can use objects and what operations they can perform on those objects. The ability to access an object is called authority. When you set up object authority, you need to be careful to give your users enough authority to do their work without giving them the authority to browse and change the system. Object authority gives permissions to the user for a specific object and can specify what the user is allowed to do with the object. You can limit an object resource through specific detailed user authorities, such as adding records or changing records. System resources can be used to give the user access to specific system-defined subsets of authorities: *ALL, *CHANGE, *USE, and *EXCLUDE.

Files, programs, libraries, and directories are the most common system objects that require resource security protection, but you can specify authority for each object on the system.

The system provides several tools to assist you in designing a straightforward resource security scheme:

Group profiles
You can group users with similar authority needs under a single group profile. Then the users in the group can all share the same authority to use objects unless you define specific authority for a user in the group.

Authorization lists
You can group objects with similar security requirements in an authorization list. Then you can grant authority to the list rather than to the individual objects. An authorization list contains a list of users and the authority that the users have to the objects that the authorization list secures. You can also use an authorization list to define public authority for the objects on the list. When you set the public authority for an object to *AUTL, the object gets its public authority from its authorization list. You cannot use an authorization list to secure a user profile or another authorization list. Also, you can specify only one authorization list for an object. For more information, see "Authorization list security" in the *iSeries Security Reference*.

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

User authority
You can define specific authority to access and use objects for individual user profiles. Using user authority may be helpful when you have a small number of users that have access requirements that do not conform to group requirements.

Object ownership
Every object on the system has an owner and the owner has *ALL authority to the object by default. Group profiles or individual user profiles can own objects. When an object has no owner or when object ownership might pose a security risk, the system assigns ownership of the object to an IBM-supplied user profile called the Default Owner (QDFTOWN) user profile. When a user creates an object, the user is the owner of the object unless the user specifies that the group profile to which the user belongs should be the owner of the object. However, you can change or remove the owner's authority to the object. Proper assignment of object ownership helps you manage applications and delegate responsibility for the security of your information. For more information, see "Object ownership" in the *iSeries Security Reference*.

Primary group authority
You can specify a primary group for an object and the authority the primary group has to the object. The system stores the name of the primary group profile and the primary group's authority for the object with the object. The use of primary group authority may simplify your authority management and improve authority checking performance in comparison to the use of group profiles. Only a user profile with a group identification number (*gid*) may be the primary group for an object and the same profile cannot be the owner of the object and its primary group. For more information, see "Group ownership of objects" in the *iSeries Security Reference*.

Library authority
Many objects on the system reside in libraries. To access an object in a library, you need authority both to the object itself and to the library in which it resides. You can put files and programs that have similar protection requirements into a library and restrict access to that library. This is often simpler than restricting access to each individual object. However, library security may not be adequate for protecting data with high security requirements. To protect highly sensitive or critical objects, you may want to secure the individual object or use an authorization list rather than rely on library security alone. For more information, see "Library security" in the *iSeries Security Reference*.

Directory authority
When you access an object in a directory, you must have authority to all the directories in the path that contains the object. You must also have the necessary authority to the object to perform the operation that you requested. 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. For example, you might choose to limit access to directories and use public authority to the objects within the directory because limiting the number of specific authorities that you define for objects improves the performance of the authority checking process.

Object authority

If the access to a library or directory is not specific enough or when specific objects within a library or directory have different access requirements than the general library or directory, you can restrict authority on individual objects, such as files. Authority to an object falls into one of three categories:

1. Object Authority defines what operations a user or program can perform on the object as a whole.
2. Data authority defines what operations a user or program can perform on the contents of the object.
3. Field authority defines what operations a user or program can perform on data fields.

For more information on object authority types, see "Defining how information can be accessed" in the *iSeries Security Reference*.

Public authority

The public consists of anyone who has authorization to sign on to the system and public authority defines what kind of access is available for users who do not have any other authority to the object. You can define public authority for every object on the system, although the public authority you define for an object may be *EXCLUDE. The system uses public authority for an object when it cannot find any other more specific authority for the object. The use of public authority is an effective means of securing objects that are not confidential and provides good system performance.

Adopted authority

Adopted authority adds the authority of a program owner to the authority of the user who runs the program. Adopted authority is a useful tool when a user needs different authority for an object in different situations. Sometimes a user needs different authorities to an object or application based on different situations in which the user works with the object or application. For example, a user might be allowed to change the information in a customer file when the user runs application programs that provide that function. However, the same user is allowed only to view, not change, customer information when the user runs a decision support tool, such as SQL. You can resolve this type of situation by giving the user *USE authority to customer information to allow file queries and use adopted authority in the customer maintenance programs to allow the user to change the files when the user runs customer maintenance programs. Objects of type *PGM, *SRVPGM, *SQLPKG, and Java™ programs can adopt authority. For more information, see "Objects that adopt the owner's authority" in the *iSeries Security Reference*.

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 authority holders is for System/36™ environment applications, which often delete program-described files and create them again. An authority holder stores the authority information for program-described database files that an application deletes and creates during processing. When a person or program deletes an object, they also delete the authority information for that object. The use of an authority holder ensures that the authority information is retained when a program deletes an object. When you delete an object, you also delete the authority information for that object. You most commonly use authority holders when you convert from the System/36 because System/36 applications often delete files and create them again. For more information, see "Authority holders" in the *iSeries Security Reference*.

Field authority

You can give field authority of Reference or Update to individual fields in a database file. The use of field authority allows you to protect the database file while allowing appropriate use of

specific fields in that file. You manage field authority only through the SQL statements GRANT and REVOKE. For more information, see "Field authority" in the *iSeries Security Reference*.

Related concepts

"Plan resource security" on page 107
This topic describes each of the components of resource security and how they all work together to protect information on your system. It also explains how to use CL commands and displays to set up resource security on your system.

"Implement resource security" on page 192
This information helps you establish resource security for workstations and printers by setting ownership and public authority to objects, as well as specific authority to applications.

System security tools

You can use security tools to manage and monitor the security environment on your system.

The security tools are part of i5/OS. They consist of a set of commands and programs that you manage through two main menus:

- The Security Tools (SECTOOLS) menu allows you to run security commands interactively.
- The Submit or Schedule Security Reports to Batch (SECBATCH) menu allows you to run security report commands in batch mode.

You can use these security tools to work with user profiles, control security auditing, print security reports, and customize your system security. For example, you can use security user profile tools to help you do the following:

- Find out what user profiles have default passwords.
- Schedule user profiles to be unavailable at certain times of the day or week.
- Schedule a user profile to be removed when the employee leaves.
- Find out which user profiles have special authorities.
- Find out who adopts authority to objects on the system.

You can the object security tools to track the public and private authorities that are associated with confidential objects. You can set these reports to print at regular intervals to help you focus your security efforts on current issues. You can also run reports to display only the changes since the last time you ran the report.

Other tools provide the ability to monitor:

- Trigger programs
- Security-relevant values in communications entries, subsystem descriptions, output queues, job queues, and job descriptions
- Altered or tampered programs

For more information on using system security tools, see "Appendix G. Commands and menus for security commands" in the *iSeries Security Reference*.

Related concepts

"Configure the system to use security tools" on page 255
This information describes how to set up your system to use the security tools that are part of i5/OS.

Security audits

This topic describes the purpose of security audits.

People audit their system security for several reasons:

- To evaluate whether the security plan is complete.

- To make sure that the planned security controls are in place and working. This type of auditing is usually performed by the security officer as part of daily security administration. It is also performed, sometimes in greater detail, as part of a periodic security review by internal or external auditors.
- To make sure that system security is keeping pace with changes to the system environment. Some examples of changes that affect security are:
  - New objects created by system users
  - New users admitted to the system
  - Change of object ownership (authorization not adjusted)
  - Change of responsibilities (user group changed)
  - Temporary authority (not timely revoked)
  - New products installed
- To prepare for a future event, such as installing a new application, moving to a higher security level, or setting up a communications network.

The techniques described here are appropriate for all these situations. Which things you audit and how often depends on the size and security needs of your organization.

Security auditing involves using commands on your system and accessing log and journal information. You can create a special profile to be used by someone doing a security audit of your system. The auditor profile needs *AUDIT special authority to change the audit characteristics of the system. Some of the auditing tasks suggested in this chapter require a user profile with *ALLOBJ and *SECADM special authority. Set the password for the auditor profile to *NONE when the audit period has ended.

For more details on security auditing, see Chapter 9, "Auditing System Security," in the *iSeries Security Reference*.

Related concepts

"Audit system values" on page 84
This topic describes the auditing system values in detail.

"Plan security auditing" on page 270
Use this information to plan security auditing for your systems.

Types of authority

This article discusses the types of authority that can be authorized and used on the server.

This system provides different types of authorities for users. **Authority** means the type of access allowed to an object. Different operations require different types of authority. For example, you might have the authority to view information or to change information on the system. The system provides several different authority types. IBM groups these authority types into categories, called **system-defined authorities** and **special authorities**.

System-defined authority to an object is divided into three categories:

Object Authority
Defines what operations can be performed on the object as a whole.

Data Authority
Defines what operations can be performed on the contents of the object.

Field Authority
defines what operations can be performed on the data fields.

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

For more information on setting up resource authority, see "How the system checks authority" in Chapter 5 of the *iSeries Security Reference*.

System-defined authorities

This table shows how system-defined authorities apply to securing files, programs, and libraries.

Use this information to plan system-defined authorities. To design simple resource security, try to plan security for entire libraries. The table shows how system-defined authorities apply to securing files, programs, and libraries:

Table 2. System-defined authorities

	***USE authority**	***CHANGE authority**	***ALL authority**	***EXCLUDE[1] authority**
Operations allowed for files	View information in the file.	View, change, and delete records in the file.	Create and delete the file. Add, change, and delete records in the file. Authorize others to use the file.	None.
Operations not allowed for files	Change or delete any information in the file. Delete the file.	Delete or clear the entire file.	None.	Any access to the file.
Operations allowed for programs	Run the program.	Change the description of the program.	Create, change, and delete the program. Authorize others to use the program.	None.
Operations not allowed for programs	Change or delete the program.	Change or delete the program.	Change the owner of the program, if the program adopts authority.	Any access to the program.
Operations allowed for libraries	• For objects in the library, any operation allowed by the authority to the specific object. • For the library, view descriptive information.	• For objects in the library, any operation allowed by the authority to the specific object. • Add new objects to the library. • Change the library description.	• Everything allowed with change authority. • Delete the library. • Authorize others to the library.	None.
Operations not allowed for libraries	• Add new objects to the library. • Change the library description. • Delete the library.	Delete the library.	None.	Any access to the library.
1 *EXCLUDE overrides any authorities that you grant to the public or through a group profile.				

Understanding how object authority and library authority work together

You also need to understand how library and object authority work together. The table below gives examples of authorities that are required for both an object and the library:

Table 3. How library authority and object authority work together

Object type	Operations	Object authority needed	Library authority needed
File	Change data	*CHANGE	*EXECUTE
File	Delete the file	*OBJOPR, *OBJEXIST	*EXECUTE
File	Create the file	None.	*EXECUTE, *ADD
Program	Run the program	*USE	*EXECUTE, *OBJOPR
Program	Recompile the program	*OBJEXIST, *OBJMGR, *READ	*ADD, *READ
Program	Delete the program	*OBJEXIST	*EXECUTE

Now you are ready to set up specific authorities for objects, directories, and libraries. For more information on the types of authorities available and some examples of how the authorities are used, see "Chapter 1. Resource Security" and "Appendix D. Authority Required for Objects Used by Commands" in the *iSeries Security Reference*.

Related concepts

"Set up specific authority for objects and libraries" on page 199
You can use the Edit Object Authority (EDTOBJAUT) command to set specific authority for the library and objects in the library.

Special authorities

This topic describes special authorities that can be specified for a user.

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.

You can specify these special authorities for a user:

***ALLOBJ**
All-object special authority allows 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 communication, and 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.

For more information on special authorities, see "Using System Security (QSecurity) System Value" in the *iSeries Security Reference*.

Related concepts

"Monitor special authorities" on page 292
This topic describes the SECBATCH menu options and commands used to monitor special authorities.

"Audit the Security Officer's actions" on page 277
A security officer or security administrator is responsible for the security on a system. A security officer has *ALLOBJ and *SECADM special authority.

Intrusion detection

Intrusion detection involves gathering information about unauthorized access attempts and attacks coming in via the TCP/IP network.

The term intrusion detection is used two ways in iSeries documentation. In the first sense, intrusion detection refers to the prevention and detection of security exposures. For example, a hacker might be trying to break into the system using an invalid user ID, or an inexperienced user with too much authority might be altering important objects in system libraries.

In the second sense, intrusion detection refers to the new intrusion detection function that uses policies to monitor suspicious traffic on the system. You can create an intrusion detection policy that audits suspicious intrusion events that come in through the TCP/IP network.

Related information

Intrusion detection

eServer Security Planner

This information describes the eServer Security Planner and explains its value.

You can use the IBM eServer Security Planner to help you plan a basic security policy for each of the operating systems that IBM servers support, including, AIX®, Linux®, i5/OS, Microsoft® Windows® 2000, and z/OS®. The planner asks you a series of questions about your business environment and your security goals. Based on your answers, the planner provides you with a list of recommendations for setting password rules, resource access rules, logging and auditing rules, and other OS specific security settings.

The planner cannot perform the configurations it suggests. Instead, the planner provides you with information and checklists to guide you as you plan and implement security on your IBM servers. In some cases, the planner also provides a program with commands you can run to apply the policy recommendations. The Security Planner now provides network security recommendations for each OS. Learn the basic concepts of designing network security including: network architecture, firewall and other network security technologies, TCP/IP security, and intrusion detection.

You need to run the planner once for each group of servers in your e-business environment that have similar security characteristics and requirements. Each usage produces a baseline security policy specific to your needs. For example, suppose that you need a very secure environment for your mission critical production system, but can tolerate more risk for your company's internal development system. Here, you would complete the planner twice, once for each system, because they each require a different level of security.

Plan your security strategy

This topic describes various aspects of planning a security strategy.

Once you have defined your company's security values within your security policy, you can begin developing your security strategy. A security strategy provides a systematic approach to all the planning tasks that are necessary for implementing your company's security policy. To best complete this goal, you need to start at the most basic security need and then work to more specific security issues.

For example, the suggested approach that this information takes is to begin with planning physical security of your hardware and information assets and then to plan specific security for your system, users, resources, and network. As you develop your security strategy, begin at the most general security concerns and then move toward other more specific security goals. Each planning step is arranged to be completed in order.

Use system values to customize your system

The system uses system values and network attributes to control many things other than security. The system and application programmers use most of these system values and attributes. The security officer should set a few system values and network attributes to customize your system.

Assign a name to your system

You use the SYSNAME network attribute to assign a name to your system. The system name appears in the upper-right corner of your sign on display and on system reports. It is also used when your system communicates with another system or with personal computers using iSeries Access for Windows.

When your system communicates with other systems or personal computers, the system name identifies and distinguishes your system from others on the network. Computers exchange system names whenever they communicate. Once you assign a system name, you should not change it, because changing it affects other systems in your network.

Choose a meaningful and unique name for your system. Even if you are not communicating with other computers today, you may in the future. If your system is part of a network, the network manager will probably tell you what system name to use.

Choose the date display format for your system

You can set the sequence in which year, month, and day appear when your system prints or displays the date. You can also specify what character the system should use between the year (Y), month (M), and day (D). The system value QDATFMT determines the date format. The following chart shows how the system prints the date, 16 June 2000, for each possible choice.

Table 4. Date and time formats

Your choice	Description	Result
YMD	Year, Month, Day	00/06/16
MDY	Month, Day, Year	06/16/00
DMY	Day, Month, Year	16/06/00
JUL	Julian Date	00/168

Note: These examples use the slash (/) date separator.

The system value QDATSEP determines what character the system uses between year, month, and day. The table below shows your choices. You use a number to specify your choice.

Table 5. Date separator characters

Separator character	QDATSEP value	Result
/ (slash)	1	16/06/00
- (hyphen)	2	16-06-00
. (period)	3	16.06.00
, (comma)	4	16,06,00
(blank)	5	16 06 00

Note: The above examples use the DMY format.

Set the time display format for your system

The QTIMSEP system value determines what character the system uses to separate hours, minutes, and seconds when it shows the time. You use a number to specify your choice. The table below shows how the time of 10:30 in the morning would be formatted using each value:

Table 6. Time separator characters

Separator character	QTIMSEP	Result
: (colon)	1	10:30:00
. (period)	2	10.30.00
, (comma)	3	10,30,00
(blank)	4	10 30 00

Decide how to name your system devices

Your system automatically configures any new display stations and printers you attach to it. The system gives a name to each new device. The QDEVNAMING system value determines how the names are assigned. The chart below shows how the system names the third display station and the second printer attached to your system:

Table 7. System device names

Your choice	Naming format	Display station name	Printer name
1	iSeries	DSP03	PRT02
2	S/36	W3	P2
3	Address of the device	DSP010003	PRT010002

Note: In the above example, the display station and printer are attached to the first cable.

Recommendations

Use naming conventions not device addresses, unless you are running software which requires S/36 naming. Names for display stations and printers are less cumbersome than names which use the address of the device. Display station and printer names appear on several Operational Assistant displays. Printer names are also used to manage printer output.

After the system has configured a new device, use the Change Display Device (CHGDEVDSP) command or the Change Printer Device (CHGDEVPRT) command to enter a meaningful description of the device. Include in the description both the physical address of the device and its location, such as John Smith's office, line 1 address 6.

Choose your system printer

Use the QPRTDEV system value to assign your system printer. This system value, the user profile, and the job description determine which printer a job uses. The job uses the system printer unless the user profile or the job description specifies a different one.

Recommendations

Normally, your system printer should be the fastest printer on your system. Use the system printer for long reports and system output.

Note: You will not know the names of your printers until you install and configure your system. Make a note about the location of your system printer now. Fill in the name of the printer later.

Allow the display of completed printer output

The system provides users the ability to find their printer output. The Work with Printer Output display shows all the output that is currently printing or waiting to print. You can also allow users to look at a list of completed printer output.

This display shows when the output printed and on what printer it printed. This can be useful in locating lost reports. The job accounting function and the QACGLVL system value allows you to display completed printer output. The *PRINT option for the QACGLVL system value allows information about completed printer output to be saved.

Storing information about completed printer output takes space on your system. Unless you think your users will print many reports, you probably do not need to provide this function. Enter NO on the System Values Selection form. This value sets the job accounting level to *NONE.

Before planning user groups

- Make sure you have written a security policy statement for your own company similar to the JKL Toy Company example that Sharon Jones and John Smith prepared.
- Make sure you have entered your choices for the system values on the System Values Selection form.
- Make notes about what you would like to include in your security memo.

After you have entered all your system options on the System Values Selection form and written a security policy, you can plan user groups.

Develop a security policy

This topic defines a security policy and explains the process for creating a security policy.

Each internet service that you use or provide poses risks to your system and the network to which it is connected. A **security policy** is a set of rules that apply to activities for the computer and communications resources that belong to an organization. These rules cover areas such as physical security, personnel security, administrative security, and network security. Your security policy defines what you want to protect and what you expect of your system users. It provides a basis for security planning when you design new applications or expand your current network. It describes user responsibilities, such as protecting confidential information and creating nontrivial passwords.

Your security policy should also describe how you will monitor the effectiveness of your security measures. Such monitoring helps you to determine whether someone might be attempting to circumvent your safeguards. To develop your security policy, you must clearly define your security objectives. Once you create a security policy, you must take steps to put into effect the rules that it contains.

You might find it useful to send security guidelines to all of your employees to emphasize your security policies regarding physical and system security. In these guidelines, you should include instructions about how to protect system security, such as signing off workstations, using passwords appropriately, and protecting the network from unauthorized intruders. The security policy could also explain the procedure for training employees and installing necessary software and hardware to ensure system security.

Remember that you can always change your security policy. When you make changes in your computing environment, you should update your security policy to address any new risks that these changes impose. Most companies find they need more strict security as they grow.

Perform the following steps to develop a security policy

1. Talk with other members of your organization, such as security auditors, to better determine your security needs.
2. Examine the technologies that you use in your company. For example, if your system is connected to the Internet, you will want a more restrictive security environment to protect your system from outside Internet users.
3. Determine your overall approach to security, as follows:

   Strict A strict policy is a need-to-know security scheme. In a strict security environment, you give users access only to the information and functions that they need to do their jobs. All others are excluded. Many auditors recommend the strict approach.

   Average
   An average security policy gives users access to objects, based on the authorities that you have assigned them.

   Relaxed
   In a relaxed security environment, you allow authorized users access to most objects on the system. You restrict access only to confidential information. A single department or small company might use the relaxed approach on their systems.
4. Determine what information assets require protection. To assist with this determination, consider confidentiality, competitiveness, and operations:

   Confidentiality
   Information that is not generally available to people in your company. Payroll is an example of confidential information. Another example of confidential information is new technical information that has not yet been announced to the public.

   Competitiveness
   Information that gives you an advantage over your competition, such as product specifications, formulas, and pricing guidelines.

   Operations
   Information on your computer that is essential for the daily operations of your business, such as customer records and inventory balances.
5. Create a statement of company policy regarding security. This is an agreement between you and the top officials in the company. Your security policy should state what your overall approach is and what assets require protection. “Example of a security policy” on page 26
6. Create a draft of your security policy. “Example: Company security memo” on page 26
7. As you work through the planning process, take additional notes that you will use to complete the security policy.
8. Complete the security policy and distribute it to the employees in your company. Use it as you implement and monitor the security on the system.

After you have created a security policy, you can choose your “Security levels” on page 6 on the system.

Example of a security policy

Overall Approach
Relaxed: Most people need access to most information.

Critical Information
- Contracts and special pricing
- Payroll (Only Accounting can set and change credit limits for customers.)
- Customer and inventory records

General Rules
- Every system user has a user profile.
- Users must change their password every 60 days.
- Users must use the latest security patches.

Figure 1. Company Security Policy

Example: Company security memo

Security of the New System

You have all attended an information meeting about our new system. Those who will use the system have started training and will begin processing customer orders next week. Observe the following security guidelines when working on your system:

- Everyone who needs to use the system will receive a user ID and a password. You will be required to change your password the first time you sign on the system and every 90 days after that. Passwords must be 8 characters in length and contain a combination of letters and numbers. Passwords must not contain your name, userid, or other personal information.
- Do not share your password with anyone. If you forget your password, go to the technical support web site for instructions on resetting your password.
- Lock your system using the screen-saver password when you are away from your desk.
- Lock up confidential information when you go home for the day. Examples of confidential information include contract and special pricing information, and payroll records.

Figure 2. Company Security Memo

Change a security policy

You can use iSeries Navigator to view and manage policies for your system.

iSeries Navigator has five policy areas:

Audit policy
This policy allows you to set up monitoring for specific actions and access to specific resources on your system.

Security policy
This policy allows you to specify the level of security and additional options that relate to system security.

Password policy
This policy allows you to specify password level security for the system.

Restore policy
This policy allows you to specify how certain objects are restored on the system.

Sign-on policy
This policy allows you to specify how user can sign onto the system.

1. From iSeries Navigator, expand your **Server → Security**.
2. Right-click **Policies** and select **Explore** to display a list of policies that you can create and manage. See iSeries Navigator help for specifics on these policies.

Plan physical security

This topic describes physical security, the key tasks for planning physical security, and explains why these tasks are important.

Physical security includes the protection of your server from accidental (or intentional) damage and theft. In addition to your server, it includes all of your workstations, printers, and storage media.

When you prepare to install your server, you should create a physical security plan by asking these questions:

- Where will you put the system unit?
- Where will you locate each display station?
- Where will you locate printers?
- What additional equipment do you need, such as wiring, telephone lines, furniture, or storage areas?
- What measures will you take to protect your system from emergencies such as fire or power interruptions?

Physical security should be part of your overall security planning. You might need special measures to protect them depending on where you put the system and its devices.

You can use the ""Physical security planning worksheet" on page 31" to record your decisions about the physical security of your system.

Plan physical security for the system unit

This topic discusses the importance of securing certain aspects of the system unit, such as the physical location, the control panel and keylock, and the Service Tools user ID and password.

Your system unit represents an important business asset and potential door into your system. Some system components inside the system are both small and valuable. You should place the system unit in a controlled location to prevent someone from stealing it or from removing valuable system components. The best location is in a private, locked room. The system unit should be in a place that can be locked before and after regular business hours.

Each system unit has a control panel that provides the ability to perform basic functions without a workstation. For example, you can use the control panel to do the following:

- Stop the system.
- Start the system.
- Load the operating system.
- Start service functions.

All of these activities can disrupt your system users. They also represent potential security exposures to your system. To prevent unauthorized use of these system operations, each system unit has either a keylock switch or an electronic keystick. They provide some protection of your system unit, but the keylock switch or the electronic keystick are not replacements for adequate physical security. To prevent the use of the control panel, place the keylock in the Secure position, remove the key, and store it in a safe place.

Risks to the system unit

In addition to theft of the system unit or its components, here are some other risks posed by inadequate physical security of your system unit:

Unintentional disruption of system operations
Many security problems come from authorized system users. Suppose that one of the display stations on your system gets locked up. The system operator is away at a meeting. The frustrated display station user walks over to the system unit, thinking that, "Maybe if I press this button, it will correct things." That button might turn off or reload the system while many jobs are running. You might need several hours to recover partially updated files. You can use the system unit keylock switch to prevent this problem from occurring.

Use of dedicated service tools (DST) function to circumvent security
Security does not control service functions the system performs, because your system software might not be operating properly when you need to perform these functions. A knowledgeable person who knows or guesses the service tools user ID and password could cause considerable damage to your system.

What to do to keep your system secure

The following information suggests ways to keep your system unit secure. Record your choices on the System Unit section of the Physical Security Planning worksheet. Also see "Example: Physical security planning form—system unit."

- Ideally, keep your system unit in a locked room. If your unit is in an unlocked room, place it where outsiders cannot access it. In addition, choose a location where responsible employees can monitor it. The following physical security features can help you protect your system from accidental or intentional tampering:
- Use the electronic keystick or the keylock:
  - Set the operating mode to Normal if you want to be able to start your system without using the key.
  - Set the operating mode to Auto if you plan to use the Automatic Power On/Off function to start and stop your system.
  - Remove the key and put it in a safe place.
- If you need to perform remote IPLs or perform remote diagnostics on your system, you might need to choose another setting for the keylock.
- Change the Service Tools (DST) user ID and password immediately after you install your system and after service personnel use it.

Example: Physical security planning form—system unit

Table 8. Physical security planning form: System unit

System unit	
Describe your security measures to protect the system unit (such as a locked room).	The system unit is in the accounting area. During the day, accounting people are always in the area and can watch the system unit. Before and after regular business hours, the area is locked.
What keylock position is normally used?	Normal.
Where is the key kept?	The key is kept in the manager's office.
Other comments relating to the system unit.	The system unit is easily accessible. The people in the accounting area should ensure that unauthorized people do not tamper with the unit.

After you plan physical security for your system unit, you can plan physical security for system documentation and storage media.

Related information

Configure service tools user IDs

Plan physical security for system documentation and storage media

This topic describes the importance of securing important system documentation and storage media. Emphasis placed on storing these items in two locations, both on-site and offsite.

System documentation includes information that IBM sends with the system, password information, your planning forms, and any reports that the system generates. Depending on your system, backup media can include tapes, CD-ROMs, diskettes, or DVD storage. You should store both system documentation and backup media at your business location as well as at another remote location. In case of a disaster, you will need this information to recover your system.

Storing system documentation securely

Service tools and security officer passwords are critical to the operation of your system. You should write these passwords down and store them in a safe, confidential location. In addition, keep a copy of these passwords at an offsite location to help you recover from a disaster.

Consider storing other important system documentation, such configuration settings and your main application libraries, away from your business location to help you recover from a disaster.

Storing your storage media securely

When you install your system, make plans for regularly saving all the information on the system to tape or other storage media. These backups allow you to recover your system if necessary. You should keep these backups in a secure location offsite as well.

Risks related to backup media and password information

- Damage to backup media: If a disaster or vandals destroyed your system backup media, you could not recover the information that was on your system, except from printed reports.
- Theft of backup media or passwords: You may have confidential business information saved on your backup media. A knowledgeable person might be able to restore this information to another computer and print or process it.

What to do to keep your storage media and passwords secure

The following information suggests ways to store your system documentation and storage media. After you have decided on your method, record your choices on the Backup Media and Documentation section of the Physical Security Planning worksheet:

- Store all passwords and backup media in a locked, fireproof cabinet.
- Take copies of your backup media to a secure, offsite location on a regular basis, for example, at least weekly.

Example: Physical security planning form—backup media and documentation

Table 9. Physical Security Planning Form: Backup Media and Documentation

Backup Media and Documentation	
Where are backup tapes stored at your business location?	`In a fireproof safe.`
Where are backup tapes stored away from your business location?	`In a fireproof safe at the office of our company's accountant.`

Table 9. Physical Security Planning Form: Backup Media and Documentation (continued)

Backup Media and Documentation	
Where are the security officer, service, and DST passwords kept?	`In the manager's office.`
Where is important system documentation, such as the serial number and the configuration, kept?	`In a fireproof safe at the office of our company's accountant.`

After you plan your storage and documentation security, you can plan physical security for your workstations.

Plan physical workstation security

This topic describes the security risks and recommendations for workstations.

You might want all users to be able to sign on at any available workstation and perform all authorized functions. However, if you have workstations that are either very public or very private, you might want to ensure that unauthorized users do not access functions on those workstations.

Risks associated with workstations

Using a workstation in a public location for unauthorized purposes
If people outside your company can easily access locations, they could potentially see confidential information. If a system user leaves a workstation signed on, someone from outside the company might be able to walk up and access confidential information.

Using a workstation in a private location for unauthorized purposes
A workstation located in a private location gives an intruder the opportunity to spend long hours trying to circumvent your security without being observed.

Using the playback function or a PC signon program on a display station to circumvent security measures
Many display stations have a record and playback function, that allows users to store frequently used keystrokes and repeat them by pressing a single key. When you use a personal computer as a workstation on the system, you can write a program to automate the signon process. Because users frequently use the signon process , they might decide to store their user IDs and passwords, rather than typing them every time they sign on.

What to do to keep your workstation secure

You need to identify which workstations might pose a security risk. The following information suggests ways to keep your workstation secure. Record your choices on the Workstations and Printers section of the Physical Security Planning worksheet. Also see "Example: Physical security planning form—workstations and printers" on page 31.

- Avoid placing workstations in very public or private locations.
- Remind users that recording a password in a display station or in a PC program violates system security.
- Require users to sign off before leaving a workstation.
- Take measures, such as using the inactive timer system values (WINACTITV and QINACTMSCQ), to prevent users from leaving workstations in public locations without signing off the system.
- Restrict access to vulnerable workstations:
  - Permit only user profiles with limited function.
  - Prevent people with security officer or service authority from signing on at every workstation using the QLMTSECOFR system value.
  - Restrict users from signing on at more than one workstation at the same time using the QLMTDEVSSN system value.

- Restrict *CHANGE authority to printers and other devices.

Example: Physical security planning form—workstations and printers

Table 10. Physical security planning form: Workstations and printers

Workstations and printers			
Workstation or printer name	Its location or description	Security exposure	Protective measures to be taken
DSP06	Loading docks	Too public	Automatic signoff. Limit functions that can be completed at the workstation.
RMT12	Remote sales office	Too private	Do not let security officer sign on there.
PRT01	Accounting office	Confidential information, such as price lists, could be seen.	Place printer in a locked room. Remind users to pick up confidential output within 30 minutes.

Plan physical security for printers and printer output

This topic describes the risks and recommendations for securing printers and printer output.

Once information starts printing, system security cannot control who sees it. To minimize the threat of someone seeing sensitive business information, you should secure printers and printer output. You should also create a policy that deals with printing confidential business information.

Risks associated with printers and printer output

When you plan security for printers, keep the following risks in mind:

- Printer location. A printer located in a public place might give unauthorized people access to confidential information.
- Printer output. Printer output left lying on a desk might reveal information.
- Confidential printer output. Employees might be printing out confidential information such as paychecks or product specifications.

What to do to keep printers and output secure

The following recommendations can help you diminish security risks that are associated with printers and their output:

- Emphasize to system users the importance of protecting confidential printer output. Include plans for protecting printers and output in your security policy.
- Avoid locating printers in public places. Consider placing the printers in a locked room.
- Schedule the printing of highly confidential output and have an authorized person stay at the printer while it prints, or require employees to pick up confidential output within a specific time interval.

Physical security planning worksheet

This topic shows the physical security planning worksheet which you can use to plan physical security of the system unit, backup media, workstations, and printers.

Table 11. Physical security planning worksheet

Physical security planning worksheet	
Prepared by:	Date:

Table 11. Physical security planning worksheet (continued)

Physical security planning worksheet	
Instructions • Learn about this worksheet in the "Plan Physical Security" topics. • Use this worksheet to describe any security issues that are related to the physical location of your system unit and attached devices. • You do not need to enter the information on this worksheet into the system.	
System unit:	
Describe your security measures to protect the system unit (such as a locked room).	
What keylock position is normally used?	
Where is the key kept?	
Other comments relating to the system unit:	
Backup media and documentation:	
Where are backup tapes stored at your business location?	
Where are backup tapes stored away from your business location?	
Where are the security officer, service, and DST passwords kept?	
Where is important system documentation, such as the serial number and the configuration, kept?	

Physical Security Planning worksheet			**Part 2 of 2**
Additional instructions for Part 2 • List below any workstations or printers whose location might cause security exposures. Indicate what protective measures you will take. For a printer, list examples of confidential printed reports under the Security Exposure column. • If you allow the system to automatically configure your local devices, you may not know the names of workstations and printers until after your system is installed. If you do not know the names when you prepare this worksheet, fill in the descriptions (such as location) and add names later.			
Physical security of workstations and printers:			
Workstation or printer name	Its location or description	Security exposure	Protective measures to be taken

Plan system security

System security entails controlling user access and their privileges, maintaining information integrity, monitoring processes and access, auditing system functions, and providing backup and recovery of security related information.

On the i5/OS, system security is integrated into the operating system through use of system values. System values control how a given function will perform based on how the value is defined. Security system values are categorized based on the functions that they perform. For example, security system values can manage the level of security on your system and signon and password controls.

Using security system values require that a user or administrator have the appropriate authority to change and update these values. In some cases, the authorities for these security values are different. For each security system value described in these sections the necessary authority is provided.

Security system values can be set through the i5/OS character-based interface or through iSeries Navigator, a graphical user interface which provides easy management of most i5/OS functions. This information provides names of system values for both iSeries Navigator and its equivalent in the character-based interface.

In addition this topic provides descriptions of these security system values, recommendations for common installations, and a form to record your system value decisions.

To complete system security planning, review these topics on security-related system values and record your choices on the System Value Selection form. See the following topics regarding security related system values:

Related concepts

"Security levels" on page 6
Security on your system is arranged in a series of levels, with each level offering a greater degree of security and protection of your data than the previous level.

General security system values

General security system values provide the cornerstone for your security policy.

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 in your security policy.

Security level system value:

This system value allows you to set the security level for the system.

The system offers five different levels of security. Each of these levels of security provide specific security controls for the system. Depending on the decisions you made in the security policy, you can select a security level that you need. IBM ships all new systems with the security level 40, which provides a high level of security that is necessary for most installations. It is not recommended that you change your security level on a new system lower that this value.

Even though IBM recommends you keep systems at level 40, lower values are described to provide a function-by-function comparison between each security level.

Table 12. Possible values for the security level system value. This table compares the different settings and the functions that the security level allows.

Security level	iSeries Navigator description	Functions allowed	Functions not allowed
10 (no security)[1]	No passwords are needed and users have authority to all resources	Provide users with *ALLOBJ access to all objects.	NA
20 (low or relaxed security)	Passwords are required and users have authority to all resources	• Provides users with *ALLOBJ access to all objects. • User name required to sign on. • Password required to sign on. • Password security active. • Menu and initial program security active. • Security auditing capabilities available. • Programs that contain restricted instructions cannot be created or recompiled. • *USRSPC, *USRIDX, and *USRQ objects can be created only in libraries specified in the QALWUSRDMN system value.	• Resource security active. • User profile created automatically. • Programs that use unsupported interfaces fail at run time. • Enhanced hardware storage protection supported. • Pointers used in parameters are validated for user domain programs running in system state. • Message handling rules are enforced between system and user state programs. • A program's associated space cannot be directly modified. • Internal control blocks are protected.
30 (medium or average security)	Passwords are required and users' access is based on their authority	• User name required to sign on. • Password required to sign on. • Password security active. • Menu and initial program security active. • Security auditing capabilities available. • Programs that contain restricted instructions cannot be created or recompiled. • *USRSPC, *USRIDX, and *USRQ objects can be created only in libraries specified in the QALWUSRDMN system value.	• Allow access to all objects. • Resource security active. • User profile created automatically. • Programs that use unsupported interfaces fail at run time. • Enhanced hardware storage protection supported. • Pointers used in parameters are validated for user domain programs running in system state. • Message handling rules are enforced between system and user state programs. • A program's associated space cannot be directly modified. • Internal control blocks are protected.

Table 12. Possible values for the security level system value (continued). This table compares the different settings and the functions that the security level allows.

Security level	iSeries Navigator description	Functions allowed	Functions not allowed
40 (high or strict security) [2]	Protect from undocumented system interfaces	• User name required to sign on. • Password required to sign on. • Password security active. • Menu and initial program security active. • Security auditing capabilities available. • Programs that contain restricted instructions cannot be created or recompiled. • *USRSPC, *USRIDX, and *USRQ objects can be created only in libraries specified in the QALWUSRDMN system value. • Pointers used in parameters are validated for user domain. • A program's associated space cannot be directly modified. • Internal control blocks are protected.	• Allow access to all objects. • User profile created automatically. • Message handling rules are enforced between system and user state programs.

Table 12. Possible values for the security level system value (continued). This table compares the different settings and the functions that the security level allows.

Security level	iSeries Navigator description	Functions allowed	Functions not allowed
50 (high or strict security) [3]	Enhance protection of system interfaces	• User name required to sign on. • Password required to sign on. • Password security active. • Menu and initial program security active. • Security auditing capabilities available. • Programs that contain restricted instructions cannot be created or recompiled. • *USRSPC, *USRIDX, and *USRQ objects can be created only in libraries specified in the QALWUSRDMN system value. • Pointers used in parameters are validated for user domain. • A program's associated space cannot be directly modified. • Internal control blocks are protected.	• Allow access to all objects. • User profile created automatically.

1. Security level 10 is no longer supported. If you change from security level 10 to 20, 30, 40 or 50, you will not be able to change it back to level 10.
2. IBM ships all new systems with a security level of 40. IBM strongly recommends that you leave the security level set to 40.
3. At security level 50, no system internal control blocks can be modified. In comparison some system internal control blocks can be modified at security level 40.

Relationship to your security policy

In your security policy, you try to maintain a balance between protecting your assets, user access, and system performance. If the system contains highly confidential material or information that would seriously compromise your business if it was lost or stolen, that system would require a higher security level than a system that contains less sensitive information. In addition, you may have a system that is connected to an insecure network, such as the Internet and could be potentially targeted for an attack. These systems also need a higher security level to protect them.

Note: Security level alone does not protect systems connected to insecure networks from attack. If you are planning to connect to the Internet or any other insecure network, you need analyze the risks not only to your system but also your entire network.

Table 13. Quick reference. Provides details for the security level system value.

iSeries Navigator name	Security level
Character-based interface name	QSECURITY
Authority	All object (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Security Policy** and select **Properties**. 3. On the **General** page, you will find the options for security level. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QSECURITY`.
Changes take effect	At next restart of the server
Default value	40 (Protect from undocumented system interfaces)
Recommended values	40 (Protect from undocumented system interfaces)
Lockable	Yes
Special considerations	If you change from security level 10 to 20, 30, 40 or 50, you will not be able to change back to level 10.

For more detailed information about this security value, see Chapter 3, ″Security System Values″ in Security Reference.

Retain server security:

This system value determines whether or not the security data needed by a server to authenticate a user on a target system through client-server interfaces can be retained on the host system.

This security value provides that you can either turn on or off this capability. However this does not include the system user profile password.

See Table 15 for an overview of the retain server security system value.

The following table describes the possible values for the retain system security value:

Table 14. Possible values for the retain server security system value

iSeries Navigator	Character-based interface	Description
Deselected	0 (No)	Server security data is not retained
Selected	1 (Yes)	Server security data is retained

Relationship to security policy

Table 15. Quick reference. Provides details for the retain server security system value.

iSeries Navigator name	Allow server security to be retained
Character-based interface name	QRETSVRSEC

Table 15. Quick reference (continued). Provides details for the retain server security system value.

iSeries Navigator name	Allow server security to be retained
Authority	All object access (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Security Policy** and select **Properties**. 3. On the **General** page, you will find the option for retaining security information. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QRETSVRSEC`.
Changes take effect	Immediately
Default value	Deselected (0)
Recommended value	
Lockable	Yes
Special considerations	If you change from allowing server security information to be retained to not allowing the information to be retained, you may cause some user applications to fail.

For more detailed information about this security value, see Chapter 3, ″Security System Values″ in Security Reference.

Share memory control:

This system value determines whether or not to allow access to shared memory, or use mapped memory stream files.

This controls how users, particularly application developers, use application programmable interfaces (APIs) that deal with sharing memory or mapped memory stream files. 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 could potentially pose a risk to your system and assets. A programmer would have write access and could add, change, and delete entries in the shared memory or stream file.

See Table 17 on page 39 for an overview of the shared memory control system value.

The following table provides a description of each of the possible settings for this system value:

Table 16. Possible values for the share memory control system value

iSeries Navigator	Character-based interface	Description
Deselected	0 (No)	Users cannot use shared memory or use mapped memory that has write capability. Setting this value prohibits users and programmers from using shared memory APIs, or mapped memory objects that have write capability. Use this value in environments with higher security requirements.
Selected	1 (Yes)	Users can use shared memory, or use mapped memory that has write capability. Setting this value allows users and programmers the ability to add, change, and delete entries in the shared memory or stream files.

Relationship to security policy

In terms of your security policy, you need to weigh the need of application performance with your need for security. If your company has applications that use shared memory you should consider allowing programmers to use these APIs. It makes application programming easier and more cost effective. However, if your environment needs stricter security, it is recommended to limit this capacity.

Table 17. Quick reference. Provides overview of the shared memory control system value.

iSeries Navigator name	Allow use of shared or mapped memory with write capability
Character-based interface name	QSHRMEMCTL
Authority	All object (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Security Policy** and select **Properties**. 3. On the **Shared Memory** page, you will find this option. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QSHRMEMCTL`.
Changes take effect	Immediately
Default value	Selected (1)
Recommended values	
Lockable	Yes

For more in-depth information about this security value, see Chapter 3, “Security System Values” in Security Reference.

Remote service attribute:

This system value allows you to analyze a system from an area other that where the system located.

By using this system value, service professionals can remotely provide problem analysis of your system and troubleshoot based on their findings. Even though it is categorized as a message and services system value, remote service attribute has security implications. If you allow remote analysis of your system, potentially any remote user could gain access to your system, if they acquired the appropriate authority.

See Table 19 for an overview of the remote service attribute system value.

Table 18. Possible values for the remote service attribute system value

iSeries Navigator	Character-based interface	Description
Deselected	0 (Off)	Remote service attribute is off
Selected	1 (On)	Remote service attribute is on

Relationship to security policy

Within your security policy, you should outline what you will do if your system needs servicing. For example, you may wish to restrict remote services until you need them. In the event that your system needs to be analyzed by a service professional, you can reset this value during the time of service then return to the original setting once all troubleshooting tasks have been completed.

Table 19. Quick Reference. Provides details for the remote server attribute system value.

iSeries Navigator name	Allow server security to be retained
Character-based interface name	QRMTSRVATR
Authority	All object (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Configuration and Service → System Values → Messages and Service**. 2. On the **Remote** page, you will find the option for the remote service attribute. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QRMTSRVATR`.
Changes take effect	Immediately
Default value	Deselected (0)
Recommended value	
Lockable	Yes
Special considerations	The recommended value will not allow anyone to perform service functions remotely. **Note:** In some situations, you may need to change this system value before you can receive assistance from a service provider or your software provider.

For more in-depth information about this security value, see Chapter 3, ″Security System Values″ in Security Reference.

Remote power-on and restart:

This system value determines whether or not you will allow remote users to power-on and restart the system.

This system value provides you the ability to start the remote system by using your telephone and a modem or the SPCN signal. This means that 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 would 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.

See Table 21 for an overview of the remote power-on and restart system value.

Table 20. Possible values for the remote power-on and restart system value

iSeries Navigator	Character-based interface	Description
Deselected	0 (No)	Do not allow remote power-on and restart
Selected	1 (Yes)	Allow remote power-on and restart

Table 21. Quick reference. Provides details for the remote power-on and restart system value.

iSeries Navigator name	Remote power-on and restart
Character-based interface name	QRMTIPL
Authority	All object (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Configuration and Service** → **System Values** → **Restart**. 2. On the **General** page, you will find the option for remote power-on and restart. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QRMTIPL`.
Changes take effect	Immediately
Default value	Deselected (0)
Recommended value	
Lockable	No
Special considerations	If you have a remote system that handles the administration of your system, you should allow remote power-on and restart.

For more detailed information about this security value, see Restart system values: Allow remote power-on and restart.

Use adopted authority:

Adopted authority adds the authority of a program owner to the authority of the user running the program.

Sometimes a user may need different authorities to an object or application. For instance, you have employees that need to update customer information by using a data management application that provides that function. However, the same users should be allowed to view, but not change, the same

customer information when using a decision support tool, such as SQL. One solution to this situation is to use adopted authority. You can use adopted authority to protect your important files from being changed outside of your approved application programs while you still allow queries against the files.

See Table 23 on page 43 for an overview of this system value.

Table 22. Possible values for the use adopted authority system value

iSeries Navigator	Character-based interface	Description
All users	*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.
Authorization list	Name of the authorization list	The user's authority is checked against the specified authorization list. This authority cannot come from adopted authority. If the user has at least the USE authority attribute in the specified authorization list, the user can create, change, or update programs or service programs that use the authority of the program which called them.

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

Relationship to security policy

This system value determines which users can work with programs with adopted authorities. Adopted authority adds the authority of a program owner to the authority of the user running the program. All users with adopted authority can create and change the program, as long as they have authority to that program. Before determining which programs and users that will use adopted authority, answer the following questions:

How much authority do users need for a given program or application?
Programs should adopt the authority of a user profile that has only enough authority to do the necessary functions, not excessive authority. You should be particularly cautious of programs that adopt the authority of a user profile that either has *ALLOBJ special authority or owns important objects. These users could have access to core program functions and alter key data or change application parameters. Adopting the authority of an application owner is preferable to adopting the authority of QSECOFR or a user with *ALLOBJ special authority. Ensure that applications owners of applications that adopt authority are not in QSECOFR user class or have *ALLOBJ special authority.

What programs should use adopted authority?
Programs that adopt authority should have a specific, limited function. Carefully monitor the function provided by programs that adopt authority. Make sure these programs do not provide a means for the user to access objects outside the control of the program, such as command entry capability. In addition programs that adopt authority should be secured properly. It is critical that you understand how a program is used before allowing adopted authority. System performance may be impacted negatively if adopted authority is used excessively. Chapter 5, ″Resource Security″ of the Security Reference book contains flowcharts that illustrate how adopted authority works.

Table 23. Quick Reference. Provides details for the use adopted authority system value.

iSeries Navigator name	**Users who can cause programs to use adopted authority from calling programs**
Character-based interface name	QUSEADPAUT
Authority	*ALLOBJ *SECADM **Note:** The QSECOFR user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Security Policy** and select **Properties**. 3. On the **General** page, you will find the option for using adopted authority. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QUSEADPAUT`.
Changes take effect	Immediately
Default value	All users
Recommended value	Authorization list
Lockable	Yes
Special considerations	This system value does not prevent anyone from creating or changing a program or service program that adopts its owner's authority. This system value applies to the Use Adopted Authority (**USEADPAUT**) parameter but not to the User Profile (**USRPRF**) parameter of a program or service program.

For more detailed information about this security value, see Chapter 3, ″Security System Values″ in Security Reference.

Allow user domain objects:

This system value specifies whether to allow user domain objects and where these objects will be located.

User domain objects can pose security risk since movement between these objects cannot be monitored. Types of user domain objects include:

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

Systems with high security requirements should restrict these user domain objects to the system's temporary library (QTEMP). Other object types, program (*PGM), server program (*SRVPGM), and SQL packages (*SQLPKG) can also be in the user domain. However, the contents of these objects cannot be changed directly and therefore are not impacted by these restrictions.

See Table 25 on page 44 for an overview of this system value.

Table 24. Possible values for the use allow user domain objects system value

iSeries Navigator	**Character-based interface**	**Description**

Table 24. Possible values for the use allow user domain objects system value (continued)

All libraries and directories	*ALL	Allows objects that are not able to be audited in all libraries and directories. The server has multiple file systems. Libraries are part of the QSYS file system and directories are part of a POSIX file system. Directories are referred to as being part of the ″root″ or ″QOpenSys″ file system.
QTEMP library and in the following: All directories	*DIR	Allows objects that are not able to be audited in all directories, in addition to the QTEMP library.
QTEMP library and in the following: Selected libraries	*library-name*	Allows you to specify libraries in which to allow objects that cannot be audited. This system value indicates specific libraries that may contain user domain versions of user objects. You may list up to 50 libraries. If you specify a list of library names, applications that currently work with user domain user objects may fail if they use objects in libraries not specified in the list.

Relationship to security policy

Table 25. Quick Reference. Provides details for the allow user domain objects system value.

iSeries Navigator name	Allow these objects in
Character-based interface name	QALWUSRDMN
Authority	*ALLOBJ *SECADM **Note:** The QSECOFR user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Security Policy** and select **Properties**. 3. On the **User Domain Objects** page, you will find the options for this system value. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QALWUSRDMN`.
Changes take effect	Immediately.
Default value	All libraries and directories.
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.
Lockable	Yes.

Table 25. Quick Reference (continued). Provides details for the allow user domain objects system value.

Special considerations	Some systems have application software that need user domain object types (*USRSPC, *USRIDX, or *USRQ). For those systems, set this system value to use a library list that includes all the libraries used by the application. All libraries that are defined with this system value, with the exception of QTEMP, should have exclude (*EXCLUDE) public authority. This limits the number of users to read or change the data in user domain objects in these libraries.

For more detailed information about this security value, see Chapter 3, ″Security System Values″ in Security Reference.

Authority for new objects:

This system value is used to determine the public authority for a newly created object.

This setting is used as the default public authority for create commands to be set throughout the system if you create a new object and do not specify an authority level.

See Table 27 for information on this system value.

Table 26. Possible values for the authority for new objects system value

iSeries Navigator	Character-based interface	Description
Change	*CHANGE	Allows the public to change newly created objects.
Use	*USE	Allows objects that are not able to be audited in all directories, in addition to the QTEMP library.
All	*ALL	Allows all users of the system, except those given an authority less than All, to completely control the newly created objects. These users will be able to read, change, delete, and manage the security of these objects.
Exclude	*EXCLUDE	The public is not allowed to use new objects.

Relationship to security policy

Table 27. Quick Reference. Provides details for the authority for new objects system value.

iSeries Navigator name	Default authority for newly created objects in QSYS.LIB file system
Character-based interface name	QCRTAUT
Authority	*ALLOBJ *SECADM **Note:** The QSECOFR user profile is shipped with these authorities.

Table 27. Quick Reference (continued). Provides details for the authority for new objects system value.

How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Security Policy** and select **Properties**. 3. On the **Public Authority** page, you will find the options for this system value. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QCRTAUT`.
Changes take effect	Immediately
Default value	Change
Recommended value	Change
Lockable	Yes
Special considerations	The authority for new objects system value is not used for objects created in directories in the enhanced file system. Several IBM-supplied libraries, including QSYS, have the create authority (CRTAUT) command set to point to the system value (*SYSVAL).

For more detailed information about this security value, see Chapter 3, ″Security System Values″ in Security Reference.

Scan file system:

This system value can be used to specify whether file systems should be scanned using the integrated file system scan-related exit programs.

You can scan for a variety of reasons, depending on how the exit program is defined. For example, you can scan for a specific text string, file name, or virus. Integrated file system scanning is enabled when exit programs are registered with any of the integrated file system scan-related exit points.

See Table 29 on page 47 for details on this system values.

Table 28. Possible values for the scan file system value

iSeries Navigator	Character-based interface	Description
Deselected	*NONE	No integrated file system objects will be scanned.
Selected	*ROOTOPNUD	Stream file objects stored in *TYPE2 [1] directories in the “root” (/), QOpenSys, and user-defined file systems will be scanned.

1. The integrated file system is comprised of several different files systems. File systems are comprised of directories which can be formatted differently. *TYPE2 directories provide enhance performance, reliability, functionality, and capacity when managing files within those directories. For more information on these directory types, see *TYPE2 directories.

Relationship to security policy

It is important to provide an explicit written statement within your security policy regarding viruses and scanning personal systems for suspicious programs. The exit programs provide security against viruses. These system values specify whether the exit programs are called.

Table 29. Quick reference. Provides details for the scan file systems system value.

iSeries Navigator name	**Use registered exit program to scan the "root" (/), QOpenSys, and user-defined file systems**
Character-based interface name	QSCANFS
Authority	*ALLOBJ *SECADM **Note:** The QSECOFR user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Security Policy** and select **Properties**. 3. On the **Scan** page, you will find the option for retaining security information. **Character-based interface** 1. In the character-based interface type `WRKSYSVAL QSCANFS`.
Changes take effect	Immediately
Default value	Selected (*ROOTOPNUD)
Lockable	Yes
Special considerations	You can provide granular control of scanned files using the options associated with scan file system control system value.

For more detailed information about this security value, see Chapter 3, ″Security System Values″ in Security Reference.

Scan file system control:

The scan file systems control 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.

This system value 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 the different scanning options and you can select to use default scan options which provide the following scan controls:

- Perform write access upgrades
- Fail close request if scan fails during close
- Scan on next access after object has been restored

See Table 31 on page 49 for details on this system value.

Optionally 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 30. Possible values for the scan file system control system value

iSeries Navigator	**Character-based interface**	**Description**
No selections	*NONE	No controls are being specified for the integrated file system scan-related exit points.

Table 30. Possible values for the scan file system control system value (continued)

iSeries Navigator	Character-based interface	Description
Scan accesses through file servers only	*FSVRONLY	Only accesses through the file servers to the system will be scanned. However, native or direct connections to the system are not scanned. If this option is not selected, all accesses will be scanned no matter if you connect directly to the system or through a file server.
Fail request if exit program fails	*ERRFAIL	This option specifies the request or operation that started the exit program will fail if there are errors when the exit program is called. If this happens, the requested operation receives an indication that the scan fail on that object. If you do not select this option, the system will skip the failing exit program and treat the object as if it was not scanned by this exit program.
Perform write access upgrades (selected) [1]	NA	This option allows the system to upgrade the access for the scan descriptor passed to the exit program to include write access, if possible. Use this option if you want the exit program to be able to fix or modify objects even though they were originally opened with read-only access.
Perform write access upgrades (deselected)	*NOWRTUPG	This option specifies that the system will not upgrade the access to include write access.
Use **only when objects have changed** attribute to control scan	*USEOCOATR	With this option, the system specifies the 'object change only' attribute to scan the object if it has been changed.
Fail close request if scan fails during close	*NOFAILCLO	This option specifies that the system will fail the close request if an object failed a scan during close processing. This option only applies to close requests. If the **Fail request if exit program fails** option is selected and this option is not selected, the system will not send a failure indication even though an object failed a scan during close processing. But, the object will be marked as failing a scan.

Table 30. Possible values for the scan file system control system value (continued)

iSeries Navigator	Character-based interface	Description
Scan on next access after object has been restored	*NOPOSTRST	This option indicates that regardless of how an object is defined with its scan attribute, the object will be scanned after it is restored. If the object scan attribute indicates that the object will not be scanned, this option forces a scan after the object is restored. If the object scan attribute indicates that the object will be scanned if it has been changed since the last scan, then the object will be scanned after a restore since the restore operation is considered a change to the object.

Relationship to security policy

Scanning control options provide granular control to using scan-related exit programs for the integrated file system. For security purposes, you can use these options to enhance detection of computer viruses and suspicious programs that may be in your integrated file system when the exit programs are designed to detect viruses.

Table 31. Quick reference. Provides details for the scan file system control system value.

iSeries Navigator name	Scan control
Character-based interface name	QSCANFSCTL
Authority	*ALLOBJ *SECADM **Note:** The QSECOFR user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Security Policy** and select **Properties**. 3. On the **Scan** page, you will find the options for scan control. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QSCANFSCTL`.
Changes take effect	Immediately
Default value	Use default scan control options

Table 31. Quick reference (continued). Provides details for the scan file system control system value.

iSeries Navigator name	Scan control
Recommended values	**For strict security environments** Select the **Fail request if exit program fails** option and ensure that the **Perform write access upgrades** is deselected. These options provide that any failures from the scan exit programs will prevent associated operations or the scan exit program from gaining additional access levels. **For less strict security environments** For most environments, you can choose not to select these options or simply use the default options.
Lockable	Yes
Special considerations	When installing code that is shipped from a trusted source, it is recommended that you specify **Scan on next access after object has been restored** during the installation.

For more detailed information about this security value, see Chapter 3, ″Security System Values″ in Security Reference.

Signon system values

You need to decide how users will sign on to the system.

Signon allows user who have authority to your system access to these resources. A signon consists of a user name and its associated password. System values control who can sign on, how users can sign on, to which devices users can sign on, and what actions the system takes when a user violates signon rules. Signon system values can be categorized into those system values that set up the signon environment, handle signon for interactive jobs, and limit signon to specified users and devices.

Signon environment

Three system values allow you to create a signon environment for your users. They provide information regarding signon activity and the number of signon attempts a users has before the system takes some action. The following system values provide ways to control the signon environment for users:

- Display signon information
- Maximum signon attempts
- Maximum signon attempts action

Interactive jobs

Interactive jobs require continual two-way communications between the user and the system to perform a task. An interactive job begins when a user signs onto a system and enters a request, and the system responds by processing the request. This pattern is repeated until the user ends the interactive job by signing off the system. These three signon related system values work together to provide security when dealing with interactive jobs:

- Time-out interval for inactive jobs
- Time-out interval action
- Time-out interval for disconnected jobs

Limiting signon

In some case you may need to limit who and what has access to your system resources. Users with all-object (*ALLOBJ) and security officer (*SECOFR) authorities may need to be limited to certain workstations and devices. Also workstations that may have physical security concerns, for example; computers that are secluded and could be used by unauthorized user to gain access to your system may also need to be limited.

- "Limit security officer" on page 61
- "Limit device sessions" on page 59
- "Remote signon control" on page 62

Display signon information:

This system value allows users to monitor attempted use of their profiles and know when a new password is needed.

This system value controls whether the user sees an informational display at signon that contains the date and time last signed on, the number of invalid signon attempts since the last signon, and the number of days until the password expire, if within 7 days of expiration.

See Table 33 for an overview of this system value.

Table 32. Possible values for the display signon information system value

iSeries Navigator	Character-based interface	Description
Deselected	0 (No)	Display is not shown.
Selected	1 (Yes)	Display is shown.

Relationship to security policy

Within your security policy you should inform users your company's expectations for managing their signon activities. The display signon information system value provides users with information regarding signon attempts and password expiration. Through the time stamp information generated with this system value, users can monitor attempts to sign on to systems. You need to document in your security policy what a user should do if they suspect inappropriate use of signon.

Table 33. Quick Reference. Provides details for the display signon information system value.

iSeries Navigator name	Display signon information
Character-based interface name	QDSPSGNINF
Authority	All object access (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Signon Policy** and select **Properties**. 3. On the **General** page, you will find the option for displaying signon information. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QDSPSGNINF`.
Changes take effect	Immediately

Table 33. Quick Reference (continued). Provides details for the display signon information system value.

iSeries Navigator name	Display signon information
Default value	Deselected (0)
Recommended value	Selected (1)
Lockable	Yes
Special considerations	NA

For more detailed information about this security value, see Chapter 3, ″Security System Values″ in Security Reference.

Maximum signon attempts:

The maximum signon attempts system value limits the number of consecutive incorrect signon attempts by local and remote users.

Incorrect signon attempts can be caused by incorrect user identification, incorrect password, or inadequate authority to the device. The maximum signon attempts system value works with the system value that specifies the action the system takes when the maximum number of signon attempts is reached. For information on this related system value, see Maximum sign on attempts action.

Some hackers may attempt to break into systems by guessing passwords. By limiting the number of signon attempts you allow, you limit their guesses. The maximum signon attempts system value determines how many signon tries you allow. Generally you want to set the value high enough to avoid frustrating users but also low enough to prevent a potential intruder too many guesses. Typically setting the value for signon attempts between 3 and 5 fulfills both of these requirements.

See Quick reference table for an overview of the maximum signon attempts system value.

Table 34. Possible values for the use maximum signon attempts system value

iSeries Navigator	Character-based interface	Description
No maximum	*NOMAX	The system allows an unlimited number of incorrect signon attempts. This value gives a potential intruder unlimited opportunities to guess a valid user ID and password combination.
Maximum number	*limit*	Specify a value from 1 through 25. The recommended number of signon attempts is three. Usually three attempts are enough to correct typing errors but low enough to help prevent unauthorized access.

Relationship to security policy

Within your security policy you should inform users your company's expectations for managing their signon activities. It is important to document the number of signon attempts that users are allowed and the action taken when that number is exceeded.

Table 35. Quick Reference. Provides details for the maximum signon attempts system value.

iSeries Navigator name	Incorrect signon attempts
Character-based interface name	QMAXSIGN

Table 35. Quick Reference (continued). Provides details for the maximum signon attempts system value.

iSeries Navigator name	Incorrect signon attempts
Authority	All object access (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Signon Policy** and select **Properties**. 3. On the **General** page, you will find the option for maximum signon attempts. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QMAXSIGN`.
Changes take effect	Immediately
Default value	3
Recommended value	3
Lockable	Yes
Special considerations	See Maximum sign on attempts action for special considerations regarding this system value.

For more detailed information about this security value, see Chapter 3, "Security System Values" in Security Reference.

Maximum signon action:

The maximum signon attempts action system value determines what the system does when the maximum number of signon attempts is reached at a workstation.

This system value works with the "Maximum signon attempts" on page 52 system value to prevent unauthorized sign on to the system.

See Table 37 on page 54 for an overview of the maximum signon action system value.

Table 36. Possible values for the maximum signon attempts system value

iSeries Navigator	Character-based interface	Description
Disable user	2	Disable user profile only.
Disable device	1	Disable device only.
Disable user and device	3	Disable both the user profile and device.

Relationship to security policy

Within your security policy you should inform users your company's expectations for managing their signon activities. It is important to document the number of signon attempts that users are allowed, and the action taken when that number is exceeded.

Table 37. Quick Reference. Provides details for the maximum signon action system value.

iSeries Navigator name	When maximum is reached
Character-based interface name	QMAXSIGNACN
Authority	All object access (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Signon Policy** and select **Properties**. 3. On the **General** page you will find the option for maximum signon attempts. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QMAXSIGNACN`.
Changes take effect	Immediately
Default value	Disable user and device (3)
Recommended value	Disable user and device (3)
Lockable	Yes
Special considerations	The recommended value for the maximum sign on attempts allows a user three consecutive attempts to sign on, by using the correct user ID and password combination. When a user exceeds the number of unsuccessful signon attempts allowed, the system will disable the user's profile and vary off the device where the user attempted to sign on. To make a user's profile available for signon again, use the following command: CHGUSRPRF USRPRF(profile-name) STATUS(*ENABLED) To make a workstation available for signon again, use the Work with Configuration Status (WRKCFGSTS) command to vary on the device.

For more detailed information about this security value, see Chapter 3, "Security System Values" in Security Reference.

Timeout interval for inactive jobs:

The timeout interval for inactive jobs system value specifies in minutes how long the system allows a job to be inactive before taking action.

A workstation is considered inactive if it is waiting at a menu or display, or if it is waiting for message input with no user interaction. User interaction includes using the Enter key, paging functions, function keys, and help functions.

The system determines which jobs are inactive. For example if a user starts a second interactive job at the same display device, then any interaction such as the pressing the Enter key on either job causes both jobs to be marked as active.

The timeout interval for inactive jobs system value works with the system value that determines what action the system takes when an inactive job exceeds the specified interval. For information of the related

system value, see Timeout interval action. When the system is started, it checks for inactive jobs that are at or exceeds the timeout interval. If a system is started at 9:30 in the morning and the timeout interval is set at 30 minutes, the system will check for inactive jobs at 10:00, 10:30, 11:00, and so on. If it discovers a job that has been inactive for 30 minutes or more, it takes the action specified by the timeout interval action system value. These two 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.

See Quick reference table for an overview of the timeout interval for inactive jobs system value.

Table 38. Possible values for the timeout interval for inactive jobs system value

iSeries Navigator	Character-based interface	Description
Do not time out	*NONE	The system does not check for inactive jobs.
5-300 (in minutes)	*interval-in-minutes*	Specify a value of 5 through 300.

Relationship to security policy

Within your security policy you should inform users your company's expectations for managing their signon activities. Inactive jobs pose a potential risk to system resources because someone could gain access to your system through an inactive terminal. However, normal job duties often interrupt users at their workstations, so you need to provide some flexibility for these expected interruptions. Using the interactive job system values provide a means to maintain security of system resource and provide user's flexibility to perform all their job responsibilities. Your security policy should state expectations for users regarding signon activities to both their workstations and to system to which they have access. For example, users are expected to password protected their workstations and enable password protection every time they leave their workstation. If they need to leave their workstation while performing a job on the system, locking their workstation provides a first barrier to anyone who tries to gain access to your system through that workstation. However, password protection is only the first line of defense. Use the interactive job system values to ensure that malicious users do not gain access to system resources.

If a job exceeds the timeout interval for inactive jobs, the system will take the specified timeout interval action. If this action is to disconnect the job, the system will wait until the timeout interval has elapsed before disconnecting the job. If this action is to end the job, the system will wait until the timeout interval has elapsed before ending the job. Assume you have set the inactive job timeout interval to 30 minutes and the inactive job action to disconnect the job. Also assume that the disconnected job timeout interval is 300 minutes or 5 hours. If a user forgets to signoff at 9:30 a.m., the system disconnects her job at 10:00 a.m. and will end the job at 3:00 p.m.

When the system ends the disconnected job, any data on the user's display that has not yet been entered into the system will be lost. If the user signs on to the same workstation before the disconnected job timeout elapses, then the job resumes from the point where the system disconnected the job.

Table 39. Quick Reference. Provides details for the timeout interval for inactive jobs system value.

iSeries Navigator name	Timeout interval
Character-based interface name	QINACTITV
Authority	All object access (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.

Table 39. Quick Reference (continued). Provides details for the timeout interval for inactive jobs system value.

iSeries Navigator name	Timeout interval
How to access	**iSeries Navigator** 1. Expand **Configuration and Service** → **System Values**. 2. Right click **Jobs** and select **Properties**. 3. On the **Interactive Jobs** page, you will find the option for timeout interval for inactive jobs. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QINACTITV`.
Changes take effect	Immediately
Default value	Do not time-out
Recommended value	60 minutes
Lockable	Yes
Special considerations	This system value is used with the timeout interval action for inactive jobs and the timeout interval for disconnected jobs system values. Together these system values ensure that inactive and disconnected jobs are ended properly.

For more in-depth information about this security value, see Chapter 3, “Security System Values” in Security Reference.

Timeout interval action:

The timeout interval action system value specifies what the system does when a job reaches the timeout interval.

If you select end the job, the system ends any job that has been inactive longer than the specified timeout interval . You can also choose to disconnect the inactive job or specify the name of a message queue, to which the system sends a warning message when a job has been inactive too long. When working with interactive jobs, this system value works with the timeout interval system value to determine the action taken after the specified time has elapsed.

See Quick reference table for an overview of the timeout interval action system value.

Table 40. Possible values for the timeout interval action system value

iSeries Navigator	Character-based interface	Description
End job	*ENDJOB	Inactive jobs are ended. If the inactive job is a group job, all jobs associated with the group are also ended. If the job is part of a secondary job, both jobs are ended.
Disconnect job	*DSCJOB	The inactive job is disconnected, as are any secondary or group jobs associated with it. If the job cannot be disconnected, the job will be ended. The disconnected job timeout interval system value controls whether the system eventually ends disconnected jobs.

Table 40. Possible values for the timeout interval action system value (continued)

iSeries Navigator	Character-based interface	Description
Send message	*message-queue-name*	Message CPI1126 is sent to the specified message queue when the inactive job timeout interval is reached. This message states: Job &3/&2/&1;has not been active .

Relationship to security policy

Within your security policy you should inform users your company's expectations for managing their signon activities. Inactive jobs pose a potential risk to system resources because someone could gain access to your system through an inactive terminal. However, normal job duties often interrupt users at their workstations, so you need to provide some flexibility for these expected interruptions. Using the interactive job system values provide a means to maintain security of system resources and provide users flexibility to perform all their job responsibilities. Your security policy should state expectations for users regarding signon activities to both their workstations and to system to which they have access. For example, users are expected to password protected their workstations and enable password protection every time they leave their workstation. If they need to leave their workstation while performing a job on the system, locking their workstation provides a first barrier to anyone who tries to gain access to your system through that workstation. However, password protection is only the first line of defense. Use the interactive job system values to ensure that malicious users do not gain access to system resources.

If a job exceeds the timeout interval for inactive jobs, the system takes the specified timeout interval action. If this action is to disconnect the job, the system also waits until the timeout interval for disconnected jobs has elapsed before ending the job. Assume that you have set the inactive job timeout to 30 minutes and the disconnected job timeout interval to 300 minutes, or 5 hours. If a user forgets to signoff at 9:30 a.m., the system disconnects that job at 10:00 a.m. and ends the job at 3:00 p.m.

When the system ends or disconnects a job, any data on the user's display that has not yet been entered into the system will be lost. If the user signs on to the same workstation before the disconnected job timeout elapses, then the job resumes from the point where the system disconnected the job.

Table 41. Quick Reference. Provides details for the timeout interval action system value.

iSeries Navigator name	When job reaches timeout
Character-based interface name	QINACTMSGQ
Authority	All object access (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Configuration and Service** → **System Values**. 2. Right click **Jobs** and select **Properties**. 3. On the **Interactive Jobs** page, you will find the option for timeout interval action . **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QINACTMSGQ`.
Changes take effect	Immediately
Default value	End job

Table 41. Quick Reference (continued). Provides details for the timeout interval action system value.

iSeries Navigator name	When job reaches timeout
Recommended value	Use disable job value unless your users run iSeries Access jobs. Disabling a job when some iSeries Access jobs are running is the equivalent of ending the jobs, which can cause significant loss of information. Use the message-queue option if you have the iSeries Access licensed program. See Chapter 8, ″Working with Messages″ of the CL Programming book shows an example of writing a program to handle messages.
Lockable	Yes
Special considerations	This system value is used with the timeout interval for inactive jobs and the timeout interval for disconnected jobs system values. Together these system values ensure that inactive and disconnected jobs are ended properly.

For more detailed information about this security value, see Chapter 3, “Security System Values” in Security Reference.

Timeout interval for disconnected jobs:

Timeout interval for disconnected jobs system value specifies how long a disconnected job can be inactive before the job is ended and the action to take when a job reaches timeout.

If you set the timeout interval action for inactive jobs to disconnect inactive jobs, you should set the timeout for disconnected jobs to end the jobs eventually.

A disconnected job uses up system resources, as well as retaining any locks on objects. To prevent two users from trying to change the same information at the same time, the system locks a record before updating it. Any locks on resources remain in effect when a the system disconnects a user’s job. Depending on your application design and the number of users on the system, locks may cause performance problems on your system. Check with your programmer or application provider to determine if locking may impact your performance.

See Quick reference table for an overview of the timeout interval for disconnected jobs system value.

Table 42. Possible values for the timeout interval for disconnected jobs system value

iSeries Navigator	Character-based interface	Description
Timeout interval (5-1,440) minutes	*time-in-minutes*	Specify a value between 5 and 1,440 minutes.
Do not timeout	*NONE	The system does not automatically end a disconnected job. To retain system performance and unlock objects, you may need to manually end jobs.
No selections	240	If you do not select either option, the system will use the default value of 240 minutes to end a disconnected job.

Relationship to security policy

Within your security policy you should inform users your company's expectations for managing their signon activities. Inactive jobs pose a potential risk to system resources because someone could gain access to your system through an inactive terminal. However, normal job duties often interrupt users at their workstations, so you need to provide some flexibility for these expected interruptions. Using the interactive job system values provide a means to maintain security of system resource and provide user's flexibility to perform all their job responsibilities. Your security policy should state expectations for users regarding signon activities to both their workstations and to system to which they have access. For example, users are expected to password protected their workstations and enable password protection every time they leave their workstation. If they need to leave their workstation while performing a job on the system, locking their workstation provides a first barrier to anyone who tries to gain access to your system through that workstation. However, password protection is only the first line of defense. Use the interactive job system values to ensure that malicious users do not gain access to system resources.

If a job exceeds the timeout interval for inactive jobs, the system will take the specified timeout interval action. If this action is to disconnect the job, the system will wait until the timeout interval has elapsed before disconnecting the job. If a job also exceeds the timeout interval for disconnected jobs, the system will end the job. Assume you have set the inactive job timeout interval to 30 minutes and the disconnected job timeout interval to 300 minutes or 5 hours. If a user forgets to signoff at 9:30 a.m., the system disconnects her job at 10:00 a.m. and will end the job at 3:00 p.m.

When the system ends the disconnected job, any data on the user's display that has not yet been entered into the system will be lost. If the user signs on to the same workstation before the disconnected job timeout elapses, then the job resumes from the point where the system disconnected the job.

Table 43. Quick Reference. Provides details for the timeout interval for disconnected jobs system value.

iSeries Navigator name	**Disconnected jobs**
Character-based interface name	QDSCJOBITV
Authority	None
How to access	**iSeries Navigator** 1. Expand **Configuration and Service** → **System Values**. 2. Right click **Jobs** and select **Properties**. 3. On the **Interactive Jobs** page, you will find the option for timeout interval for disconnected jobs. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QDSCJOBITV`.
Changes take effect	Immediately
Default value	240
Recommended value	300
Lockable	Yes
Special considerations	This system value is used with the timeout interval for inactive jobs and the timeout interval action system values. Together these system values ensure that inactive and disconnected jobs are ended properly.

For more in-depth information about this security value, see Chapter 3, "Security System Values" in Security Reference.

Limit device sessions:

The limit device sessions system value specifies whether a user is allowed to be signed on to more than one device at a time.

This value does not restrict the System Request menu or a second signon 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. Allowing users to sign on to only one workstation at a time promotes good security habits. If you limit users to one device, you discourage users from sharing user IDs and passwords. If people share user IDs, you lose both control and accountability. You can no longer tell who really does what functions on the system. In addition users must remember to sign off one workstation before moving to another one. Workstations left signed on, but not in use, pose a security risk. Give every system user a unique user ID and password with the appropriate authorities, then restrict them to using one workstation at a time. You can also restrict users to a specific device through individual user profiles.

See Table 45 table for an overview of the limit device sessions system value.

Table 44. Possible values for the limit device sessions system value

iSeries Navigator	Character-based interface	Description
Deselected	0 (No)	The system allows an unlimited number of signon sessions.
Selected	1 (Yes)	Users are limited to one device session.

Relationship to security policy

Setting the limit device sessions system value discourages sharing password and leaving workstation signed on; however, regardless of the decision you make for this system value, your security policy should implicitly discourage these practices. These bad habits provide a potential attacker access to your resources and sensitive business information. In your security policy users should be made aware of the risks and the consequences for these practices.

Table 45. Quick Reference. Provides details for the limit device sessions system value.

iSeries Navigator name	Limit each user to one device session
Character-based interface name	QLMTDEVSSN
Authority	All object access (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Signon Policy** and select **Properties**. 3. On the **General** page, you will find the option for limiting device sessions. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QLMTDEVSSN`.
Changes take effect	Immediately
Default value	Deselected
Recommended value	Selected
Lockable	Yes
Special considerations	NA

For more detailed information about this security value, see Chapter 3, “Security System Values” in Security Reference.

Limit security officer:

You may want to restrict users with authority to change security and control objects to certain workstations.

This prevents these users from signing on to workstations in remote locations without your knowledge. The limit security officer 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. This system value restricts the security officer, users with authority over all the objects on the system, and service personnel to the console. To give these users access to other devices, you can use the (GRTOBJAUT) command.

See Quick reference table for an overview of the limit security officer system value.

Table 46. Possible values for the limit security officer system value

iSeries Navigator	Character-based interface	Description
Deselected	0 (No)	Users with *ALLOBJ or *SERVICE special authority can sign on at any display station for which they have change (*CHANGE) authority. They can receive *CHANGE authority through private or public authority or because they have *ALLOBJ special authority.
Selected	1 (Yes)	A user with *ALLOBJ or *SERVICE special authority can sign on at a display station only if that user is specifically authorized (that is, given *CHANGE authority) to the display station or if user profile QSECOFR is authorized (given *CHANGE authority) to the display station. This authority can not come from public authority.

Relationship to security policy

Limiting the workstation access that users with *ALLOBJ and *SERVICE special authorities allows you to monitor the activities that these users perform. You can monitor their access on these devises and react to any suspicious activity quickly. You security policy should document which devices will be used by these users.

Table 47. Quick Reference. Provides details for the limit security officer system value.

iSeries Navigator name	Restrict privileged users to specific devices
Character-based interface name	QLMTSECOFR
Authority	All object access (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.

Table 47. Quick Reference (continued). Provides details for the limit security officer system value.

iSeries Navigator name	Restrict privileged users to specific devices
How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Signon Policy** and select **Properties**. 3. On the **General** page, you will find the option for limiting privileged users. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QLMTSECOFR`.
Changes take effect	Immediately
Default value	Deselected
Recommended value	Always display signon
Lockable	Yes
Special considerations	In order for the limit security officer system value to work, your system security level needs to be 30 or higher.

For more detailed information about this security value, see Chapter 3, "Security System Values" in Security Reference.

Remote signon control:

The remote signon control system value determines whether your system will require users to sign on when they request a passthrough or Telnet session from another server.

See Quick reference table for an overview of the remote signon control system value.

Table 48. Possible values for the remote signon control system value

iSeries Navigator	Character-based interface	Description
Always display signon	*FRCSIGNON	Remote signon requests must go through the normal signon process.
Source and target user IDs must match	*SAMEPRF	When the source and target user profile names are the same, the signon display may be bypassed if automatic signon 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 signon attempt, the pass-through session always ends and an error message is sent to the user. However, if the profile names are different, this value indicates that the session ends with a security failure even if the user entered a valid password for the remote user profile.

Table 48. Possible values for the remote signon control system value (continued)

iSeries Navigator	Character-based interface	Description
Verify user ID on target system	*VERIFY	This value allows you to bypass the signon display of the target system if valid security information is sent with the automatic signon request. If the password is not valid for the specified target user profile, the pass-through session ends with a security failure.
Reject remote signons	*REJECT	No remote signon is permitted. For TELNET access, no action is taken if this value is specified.
Invoke user-written exit program	*program-name library-name*	The program specified runs at the start and end of every pass-through session.

Relationship to security policy

For your security policy you need to know how users and systems require access to resources before determining the setting for this security value. For instance, if your employees use iSeries Access for Windows, it is recommended that you set this system value to require normal signon procedures or force that signon on both the source and target systems be the same. For user who do not use iSeries Access, you can reject remote signon.

Table 49. Quick Reference. Provides details for the remote signon control system value.

iSeries Navigator name	Remote signon
Character-based interface name	QRMTSIGN
Authority	All object access (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Signon Policy** and select **Properties**. 3. On the **Remote** page, you will find the option for remote signon control. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QRMTSIGN`.
Changes take effect	Immediately
Default value	Deselected
Recommended value	Selected
Lockable	Yes
Special considerations	If you do not want to allow any pass-through or access to iSeries Access, set this value to reject all remote signons.

For more in-depth information about this security value, see Chapter 3, “Security System Values” in Security Reference.

Password system values

In addition to setting signon system values, you also need to decide rules regarding users passwords

Password system values allow you to customize password rules that fit with your overall security environment. In general these values support the basic password requirements that you have documented in your security policy. The following information provides information on each of these system values:

Set password rules:

Follow these steps to secure your system signon.

You need to do this first.

- Set a policy that states that passwords must not be trivial and must not be shared.
- Set system values to help you with enforcement. Table 1 shows recommended system value settings.

The combination of values in table is fairly restrictive and is intended to significantly reduce the likelihood of trivial passwords. However, your users may find it difficult and frustrating to select a password that meets these restrictions.

Consider providing users with the following:

1. A list of the criteria for passwords.
2. Examples of passwords that are and are not valid.
3. Suggestions for how to think of a good password.

Use the Print System Security Attributes (PRTSYSSECA) command to print your current settings for these system values.

Table 50. System values for passwords

System value name	Description	Recommended value
QPWDEXPITV	How often the system users must change their passwords. You can specify a different value for individual users in the user profile.	60 (days)
QPWDLMTAJC	Whether the system prevents adjacent characters that are the same.	1 (yes)
QPWDLMTCHR	What characters may not be used in passwords.	AEIOU#$@
QPWDLMTREP	Whether the system prevents the same character from appearing more than once in the password.	2 (not allowed consecutively)
QPWDLVL	Whether user profile passwords are limited to 10 characters or a maximum of 128.	0[2]
QPWDMAXLEN	The maximum number of characters in a password.	8
QPWDMINLEN	The minimum number of characters in a password.	6
QPWDPOSDIF	Whether each character in a password must be different from the character in the same position on the previous password.	1 (yes)

Table 50. System values for passwords (continued)

System value name	Description	Recommended value
QPWDRQDDGT	Whether the password must have at least one numeric character.	1 (yes)
QPWDRQDDIF	How long a user must wait before using the same password again.	5 or less (expiration intervals)[1]
QPWDVLDPGM	What exit program is called to validate a newly assigned password.	*NONE

Note:

1. The QPWDEXPITV system value specifies how often you must change your password, such as every 60 days. This is the expiration interval. The QPWDRQDDIF system value specifies the number of time a user must change their password before then can use their original password again.
2. QPWDLMTCHR is not enforced at password levels 2 or 3.

Password level:

This system value allows you to set a specific password environment where all user profile passwords can have the same length specification.

You can set the password level so that passwords can be shorter, from 1-10 characters, or longer passwords from 1-128 characters. The password level can be set to allow a passphrase as the password value. A passphrase describes a password value which can be very long and has few, if any, restrictions on the characters used in the password value. You can create passphrase that contain blanks between letters, which allows you to have a sentence or sentence fragments for password values. The only restrictions on a passphrase are that it cannot start with an asterisk ('*') and trailing blanks will be removed.

See Quick reference table for an overview of the password level system value.

Table 51. Possible values for the use password level system value

iSeries Navigator	Character-based interface	Description
Short passwords using a limited character set. (0)	0	Password level 0 supports passwords that contain 1-10 alphanumeric characters as well as, $, @, #, and _. Use password level 0 if your system communicates with other servers in a network and those servers either use password level 0 passwords or run on pre-V5R1 versions of the operating system.
Short passwords using a limited character set. Disable NetServer™ passwords for Windows 95/98/ME clients. (1)	1	Password level 1 supports the same character set as password level 0, but provides improved security because it removes all NetServer[1] passwords from the system. If you require iSeries NetServer, set your password level to 0 or 2 instead.

Table 51. Possible values for the use password level system value (continued)

iSeries Navigator	Character-based interface	Description
Long passwords using an unlimited character set. (2)	2	Password level 2 supports passwords that contain 1-128 characters, and are case sensitive. You can use password level 2 if your system communicates with iSeries NetServer and all user passwords are 1-14 characters long. However, do not use password level 2 if your system communicates with other systems that use password level 0 or 1 passwords or run on pre-V5R1 versions of the operating system.
Long passwords using an unlimited character set. Disable iSeries NetServer passwords for Windows 95/98/ME clients. (3)	3	Password level 3 supports passwords that contain 1-128 characters, and are case sensitive. You cannot use this level when your system communicates with: • Other systems in a network and those systems are running with either a password level of 0 or 1 • systems that are running an operating system release less than V5R1M0 of OS/400®. • Any other system that limits the length of passwords from 1-10 characters. • The iSeries Support for Windows Network Neighborhood (iSeries NetServer) [1] product. • PCs that are using versions of iSeries Access that are V5R1 or earlier of OS/400.

1. The NetServer product for Windows 95/98/ME will not connect to the system when the password level is set to 1 or 3. NetServer passwords are removed from the system at these password levels because of security concerns with the weak encryption used for NetServer passwords. The passwords are easy to decode.

Relationship to security policy

These options provide flexibility in password security based on your security environment. Shorter passwords provide users with easier password management, since there is less chance for misspelling or forgetting password sequences; however, shorter passwords with specific password rules can be guessed by a potential hacker. A longer more involved passwords or passphrases are harder to guess, but can frustrate users and make password management more difficult. For strict security environments you may want to provide passwords that are longer, but provide suggestions for aiding users to remember these passwords. Suggest that users create passphrases that are based on something personal that they can remember easily.

For security environments that have less strict requirements, you can choose a password level that allows for shorter passwords and provide specific rules of password conduct. Whatever password level you choose, provide examples of valid password values and suggestions for formulating original passwords and passphrases. Stress that the passwords provided in your security policy are merely examples and should never be used for actual password values.

Table 52. Quick Reference. Provides details for the password level system value.

iSeries Navigator name	**Password level (at next restart)**
Character-based interface name	QPWDLVL
Authority	All object access (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Password Policy** and select **Properties**. 3. On the **General** page, you will find the options for password level. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QPWDLVL`.
Changes take effect	At next restart
Default value	Short passwords using a limited character set (0)
Recommended value	See Special Considerations
Lockable	Yes
Special considerations	**Changing password levels** You cannot change password level 3 to 0 or 1. Since all passwords used at password level 0 or 1 are removed from the system when you change to the password level 3, you must first change the password level from 3 to 2 and then to 1 or 0. At password level 2, you must change all user profile passwords to comply with the character length specified for password level 0 or 1 (10 or less characters) prior to changing to password level 1 or 0. Otherwise, users will not be able to sign on to your system. After changing these passwords you can verify that user profiles to ensure their password comply with the password level to which you are changing. See the online help for Password level for instructions. For detailed considerations and for changing password level, see the section about planning password level changes in Security Reference.

For more in-depth information about this security value, see Chapter 3, ″Security System Values″ in Security Reference.

Plan password level changes:

Operations with other systems may fail or users may not be able to sign on to the system if you haven't planned for the password level change adequately.

Changing password levels should be planned carefully. Prior to changing the QPWDLVL system value, make sure you have saved your security data using the SAVSECDTA or SAVSYS command. If you have a current backup, you will be able to reset the passwords for all users' profiles if you need to return to a lower password level.

Products that you use on the system and on clients with which the system interfaces, may have problems when the password level (QPWDLVL) system value is set to 2 or 3. Any product or client that sends passwords to the system in an encrypted form, rather than in the clear text a user enters on a signon screen, must be upgraded to work with the new password encryption rules for QPWDLVL 2 or 3. Sending the encrypted password is known as password substitution.

Password substitution is used to prevent a password from being captured during transmission over a network. Password substitutions generated by older clients that do not support the new algorithm for QPWDLVL 2 or 3, even if the specific characters are correct, will not be accepted. This also applies to any iSeries to iSeries peer access which utilizes the encrypted values to authenticate from one system to another.

The problem is compounded by the fact that some affected products, such as Java Toolbox, are provided as middle-ware. 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 middle-ware. Given this and other scenarios, it is easy to see why careful planning is necessary before changing the QPWDLVL system value.

Considerations for changing QPWDLVL from 0 to 1:

Keep these items in mind as you consider changing your password level.

Password level 1 allows a system, which does not have a need to communicate with the Windows 95/98/ME or AS/400® Client Support for Windows Network Neighborhood (iSeries NetServer) product, to have the iSeries NetServer passwords eliminated from the system. Eliminating unnecessary encrypted passwords from the system increases the overall security of the system.

At QPWDLVL 1, all current, pre-V5R1 password substitution and password authentication mechanisms will continue to work. There is very little potential for breakage except for functions and services that require the iSeries NetServer password.

Considerations for changing QPWDLVL from 0 or 1 to 2:

Password level 2 introduces the use of case sensitive passwords up to 128 characters in length, also called passphrases, and provides the maximum ability to revert back to QPWDLVL 0 or 1.

Regardless of the password level of the system, password level 2 and 3 passwords are created whenever a password is changed or a user signs on to the system. Having a level 2 and 3 password created while the system is still at password level 0 or 1 helps prepare for the change to password level 2 or 3.

Prior to changing QPWDLVL to 2, you should use the DSPAUTUSR or PRTUSRPRF TYPE(*PWDINFO) commands to locate all user profiles which do not have a password that is usable at password level 2. Depending on which profiles these commands locate, you may want to use one of the following mechanisms to have a password level 2 and 3 password added to the profiles.

- Change the password for the user profile using the CHGUSRPRF or CHGPWD CL command or the QSYCHGPW API. This will cause the system to change the password that is usable at password levels 0 and 1; and the system also creates two equivalent case sensitive passwords that are usable at password levels 2 and 3. An all uppercase and all lowercase version of the password is created for use at password level 2 or 3.

  For example, changing the password to C4D2RB4Y results in the system generating C4D2RB4Y and c4d2rb4y password level 2 passwords.
- Sign on to the system through a mechanism that presents the password in clear text, not using password substitution. If the password is valid and the user profile does not have a password that is usable at password levels 2 and 3, the system creates two equivalent case sensitive passwords that are usable at password levels 2 and 3. An all uppercase and all lowercase version of the password is created for use at password level 2 or 3.

The absence of a password that is usable at password level 2 or 3 can be a problem whenever the user profile also does not have a password that is usable at password levels 0 and 1 or when the user tries to sign on through a product that uses password substitution. In these cases, the user will not be able to sign on when the password level is changed to 2.

If a user profile does not have a password that is usable at password levels 2 and 3, the user profile does have a password that is usable at password levels 0 and 1, and the user signs on through a product that sends clear text passwords, then the system validates the user against the password level 0 password and creates two password level 2 passwords (as described above) for the user profile. Subsequent signons will be validated against the password level 2 passwords.

Any client or service which uses password substitution will not work correctly at QPWDLVL 2 if the client or service hasn't been updated to use the new password or passphrase substitution scheme. The administrator should check whether a client or service which hasn't been updated to the new password substitution scheme is required.

The clients and services that use password substitution include:

- TELNET
- iSeries Access
- iSeries Host Servers
- QFileSrv.400
- iSeries NetServer print support
- DDM
- DRDA®
- SNA LU6.2

It is highly recommended that the security data be saved prior to changing to QPWDLVL 2. Having a backup of your security data can help make the transition back to QPWDLVL 0 or 1 easier if that becomes necessary.

It is recommended that the other password system values, such as QPWDMINLEN and QPWDMAXLEN, not be changed until after some testing at QPWDLVL 2 has occurred. This will make it easier to transition back to QPWDLVL 1 or 0 if necessary. However, the QPWDVLDPGM system value must specify either *REGFAC or *NONE before the system will allow QPWDLVL to be changed to 2.

Therefore, if you use a password validation program, you may wish to write a new one that can be registered for the QIBM_QSY_VLD_PASSWRD exit point by using the ADDEXITPGM command.

iSeries NetServer passwords are still supported at QPWDLVL 2, so any function or service that requires an iSeries NetServer password should still work correctly. Once the administrator is comfortable with running the system at QPWDLVL 2, they can begin to change the password system values to exploit longer passwords. However, the administrator needs to be aware that longer passwords will have these effects:

- If passwords greater than 10 characters are specified, the password level 0 and 1 password is cleared. This user profile would not be able to signon if the system is returned to password level 0 or 1.
- If passwords contain special characters or do not follow the composition rules for simple object names (excluding case sensitivity), the password level 0 and 1 password is cleared.
- If passwords greater than 14 characters are specified, the iSeries NetServer password for the user profile is cleared.
- The password system values only apply to the new password level 2 value and do not apply to the system generated password level 0 and 1 password or iSeries NetServer password values (if generated).

Considerations for changing QPWDLVL from 2 to 3:

Keep these items in mind as you consider changing your password level.

After running the system at QPWDLVL 2 for some period of time, the administrator can consider moving to QPWDLVL 3 to maximize his password security protection.

At QPWDLVL 3, all iSeries NetServer passwords are cleared so a system should not be moved to QPWDLVL 3 until there is no need to use iSeries NetServer passwords.

At QPWDLVL 3, all password level 0 and 1 passwords are cleared. The administrator can use the DSPAUTUSR or PRTUSRPRF commands to locate user profiles which don't have password level 2 or 3 passwords associated with them.

Change known passwords:

Do the following to close some well-known entrances into the server that may exist on your system.

You will need information from these tables for some of the steps in this procedure.

Table 53. Passwords for IBM-supplied profiles

User ID	Password	Recommended value
QSECOFR	QSECOFR[1]	A nontrivial value known only to the security administrator. Write down the password that you have selected and store it in a safe place.
QSYSOPR	QSYSOPR	*NONE[2]
QPGMR	QPGMR	*NONE[2]
QUSER	QUSER	*NONE[2, 3]
QSRV	QSRV	*NONE[2]
QSRVBAS	QSRVBAS	*NONE[2]

Note:

1. The system arrives with the Set password to expired value for the QSECOFR set to *YES. The first time that you sign on to a new system, you must change the QSECOFR password.
2. The system needs these user profiles for system functions, but you should not allow users to sign on with these profiles. For new systems installed with V3R1 or later releases, this password is shipped as *NONE. If you run the CFGSYSSEC command, the system sets these passwords to *NONE.
3. To run iSeries Access for Windows using TCP/IP, the QUSER user profile must be enabled.

Table 54. Passwords for dedicated service tools

DST Level1	User ID[1]	Password	Recommended value
Basic capability	11111111	11111111	A nontrivial value known only to the security administrator.[2]
Full capability	22222222	22222222[3]	A nontrivial value known only to the security administrator.[2]
Security capability	QSECOFR	QSECOFR[3]	A nontrivial value known only to the security administrator.[2]

Table 54. Passwords for dedicated service tools (continued)

DST Level1	User ID[1]	Password	Recommended value
Service capability	QSRV	QSRV[3]	A nontrivial value known only to the security administrator.[2]

Note:

1. A user ID is only required for PowerPC® AS (RISC) releases of the operating system.
2. If your hardware service representative needs to sign on with this user ID and password, change the password to a new value after the hardware service representative leaves.
3. The service tools user profile will expire as soon as it is used for the first time.

1. Make sure that no user profiles still have default passwords (equal to the user profile name). You can use the Analyze Default Passwords (ANZDFTPWD) command.
2. Try to sign on to your system with the combinations of user profiles and passwords that are shown in the table, "Passwords for IBM-supplied profiles." These passwords are published, and they are the first choice of anyone who is trying to break into your system. If you can sign on, use the Change User Profile (CHGUSRPRF) command to change the password to the recommended value.
3. Start the Dedicated Service Tools (DST) and try to sign on with the passwords that are shown in Table 2.
4. If you can sign on to DST with any of these passwords, you should change the passwords. DST passwords can only be changed by an authenticated device. This is also true for all passwords and corresponding user IDs that are identical. For more information on authenticated devices, see the Operations Console setup information.
5. Finally, make sure that you cannot sign on just by pressing the Enter key at the Sign On display without entering a user ID and password. Try several different displays. If you can sign on without entering information on the Sign On display, do one of the following:
   a. Change to security level 40 or 50 (QSECURITY system value). Your applications might run differently when you increase your security level to 40 or 50.
   a. Change all of the workstation entries for interactive subsystems to point to job descriptions that specify USER(*RQD).

Avoid default passwords:

When you create a new user profile, the default is to make the password the same as the user profile name.

Default passwords provide an opportunity for someone to enter your system, if someone knows your policy for assigning profile names and knows that a new person is joining your organization.

When you create new user profiles, consider assigning a unique, non-trivial password instead of using the default password. Tell the new user the password confidentially, such as in a “Welcome to the System” letter that outlines your security policies. Require the user to change the password the first time that the user signs on by setting the user profile to PWDEXP(*YES).

You can use the Analyze Default Passwords (ANZDFTPWD) command to check all the user profiles on your system for default passwords. When you print the report, you have the option of specifying that the system should take action (such as disabling the user profile) if the password is the same as the user profile name. The ANZDFTPWD command prints a list of the profiles that it found and any action that it took.

Note: Passwords are stored on your system in one-way encrypted form. They cannot be decrypted. The system encrypts the specified password and compares it to the stored password just as it would check a password when you sign on to the system. If you are auditing authority failures

(*AUTFAIL), the system will write a PW audit journal entry for each user profile that does not have a default password (for systems running V4R1 or earlier releases). Beginning with V4R2, the system does not write PW audit journal entries when you run the ANZDFTPWD command.

Change to a lower password level:

There are considerations for you to make before you change to a lower password level.

Returning to a lower QPWDLVL value, while possible, is not going to be a completely painless operation. In general, the mind set should be that changing from lower QPWDLVL values to higher QPWDLVL values is a one-way trip. However, there may be cases where a lower QPWDLVL value must be reinstated.

The following sections each discuss the work required to move back to a lower password level.

Considerations for changing from QPWDLVL 3 to 2

This change is relatively easy. Once the QPWDLVL is set to 2, the administrator needs to determine if any user profile is required to contain iSeries NetServer passwords or password level 0 or 1 passwords and, if so, change the password of the user profile to an allowable value.

Additionally, the password system values may have to be changed back to values compatible with iSeries NetServer and password level 0 or 1 passwords, if those passwords are needed.

Considerations for changing from QPWDLVL 3 to 1 or 0

Because of the very high potential for causing problems for the system, such as no one can being able to sign on because all of the password level 0 and 1 passwords have been cleared, this change is not supported directly. To change from QPWDLVL 3 to QPWDLVL 1 or 0, the system must first make the intermediary change to QPWDLVL 2.

Considerations for changing from QPWDLVL 2 to 1

Prior to changing QPWDLVL to 1, the administrator should use the DSPAUTUSR or PRTUSRPRF TYPE(*PWDINFO) commands to locate any user profiles that do not have a password level 0 or 1 password. If the user profile will require a password after the QPWDLVL is changed, the administrator should ensure that a password level 0 and 1 password is created for the profile using one of the following mechanisms:

- Change the password for the user profile using the CHGUSRPRF or CHGPWD CL command or the QSYCHGPW API. This will cause the system to change the password that is usable at password levels 2 and 3; and the system also creates an equivalent uppercase password that is usable at password levels 0 and 1. The system is only able to create the password level 0 and 1 password if the following conditions are met:
  - The password is 10 characters or less in length.
  - The password can be converted to uppercase EBCDIC characters A-Z, 0-9, @, #, $, and underscore.
  - The password does not begin with a numeric or underscore character.

  For example, changing the password to a value of RainyDay would result in the system generating a password level 0 and 1 password of RAINYDAY. But changing the the password value to Rainy Days In April would cause the system to clear the password level 0 and 1 password, as the password is too long and it contains blanks. No message or indication is produced if the password level 0 or 1 password could not be created.
- Sign on to the system through a mechanism that presents the password in clear text (does not use password substitution). If the password is valid and the user profile does not have a password that is

usable at password levels 0 and 1, the system creates an equivalent uppercase password that is usable at password levels 0 and 1. The system is only able to create the password level 0 and 1 password if the conditions listed above are met.

The administrator can then change QPWDLVL to 1. All iSeries NetServer passwords are cleared when the change to QPWDLVL 1 takes effect (next IPL).

Considerations for changing from QPWDLVL 2 to 0

The considerations are the same as for changing from QPWDLVL 2 to 1 except that all iSeries NetServer passwords are retained when the change takes effect.

Considerations for changing from QPWDLVL 1 to 0

After changing QPWDLVL to 0, the administrator should use the DSPAUTUSR or PRTUSRPRF commands to locate any user profiles that do not have an iSeries NetServer password. If the user profile requires an iSeries NetServer password, it can be created by changing the user's password or signing on through a mechanism that presents the password in clear text. The administrator can then change QPWDLVL to 0.

Password expiration interval:

The password expiration interval 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. You can set this value globally for all user profiles on the system or customize the password expiration for individual user profiles. For example you may want the security officer or other users with all object (*ALLOBJ) special authority to change their passwords more frequently than the rest of your users.

See Quick reference table for an overview of the password expiration interval system value.

Table 55. Possible values for the password expiration interval system value

iSeries Navigator	Character-based interface	Description
Never Expire	*NOMAX	Users are not required to change their passwords.
Days after last change (1-366)	*limit-in-days*	Specify the number of days a password is valid before it expires.

Relationship to security policy

Within your security policy, you should describe the password rules that are defined by the system values-related passwords. For this system value, let users know how long passwords on the system are valid and what they are required to do when the expiration date is exceeded. Several other password system values force users to make unique password every time their passwords expire on the system. Be sure to document those rules as well in your security policy.

Stricter security environments would benefit from a shorter interval for password expiration. User should change their passwords periodically. This discourages sharing passwords with other system users. Passwords with a long or indefinite expiration interval provide potential intruders a longer period of access if they steal or obtain a password to a system. If an intruder obtained a valid password, potentially they could do damage or steal vital data on your system over a long period of time. If the expiration interval is shorter, then intruders would be limited in the amount of time they had access to your system. However, valid users may become frustrated if they are asked to change passwords too frequently. To

strike a balance between protection and user needs, select a value between 30 and 90 days. For most installations that range is adequate. You may need to customize password expiration for individual users or systems. Perhaps you want your security administrator or any users with all object (*ALLOBJ) authority to change passwords more frequently to minimize the threat of someone stealing those passwords. You also may want to have shorter or longer password expiration intervals for specific systems, depending on the data that these systems contain.

Table 56. Quick Reference. Provides details for the password expiration interval system value.

iSeries Navigator name	**Expiration**
Character-based interface name	QPWDEXPITV
Authority	All object access (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Password Policy** and select **Properties**. 3. On the **Expiration** page, you will find the options for password expiration. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QPWDEXPITV`.
Changes take effect	Immediately
Default value	Never expire
Recommended value	From 30 to 90 days
Lockable	Yes
Special considerations	NA

For more in-depth information about this security value, see Chapter 3, “Security System Values” in Security Reference.

Minimum length of passwords:

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

The possible values vary depending on the password level for your system. If your password level is either 0 or 1, then the possible values for minimum length are 1 through 10. If your password level 2 or 3, then the possible values for minimum length are 1 through 128.

See Quick reference table for an overview of the minimum length of passwords system value.

Table 57. Possible values for the minimum length of passwords system value

iSeries Navigator	**Character-based interface**	**Description**
Minimum length	*minimum-number-of-characters*	Specify the minimum number of characters for a password.

Relationship to security policy

Within your security policy you should describe the password rules that are defined by the system values related passwords. For this system value, let users know the minimum number of characters that a valid password contains. This system value works with the maximum length of password system value to

create a range for the length of passwords. Valid passwords must be within that length range. Remember that the length of passwords is dependant on the system. Although you may not want to disclose the password level to system users, you will want to document the length of passwords that the password level allows. For example, if you set your password level at 3, users would need to know that their passwords could be in a range of 1 to 128 characters.

Table 58. Quick Reference. Provides details for the minimum length of passwords system value.

iSeries Navigator name	**Minimum length**
Character-based interface name	QPWDMINLEN
Authority	All object access (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Password Policy** and select **Properties**. 3. On the **Validation** page, you will find the options for password length. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QPWDMINLEN`.
Changes take effect	Immediately
Default value	6
Recommended value	This value depends on what you selected for password level.
Lockable	Yes
Special considerations	NA

For more in-depth information about this security value, see Chapter 3, ″Security System Values″ in Security Reference.

Maximum length of passwords:

This system value controls the maximum number of characters in a password.

Controlling the maximum length of passwords provides additional security by preventing users from specifying passwords that are too long and have to be recorded somewhere because they cannot be easily remembered. Some communications networks require a password that is 8 characters or less. Use this system value to ensure that passwords meet the requirements of your network.

See Quick reference table for an overview of the maximum length of passwords system value.

Table 59. Possible values for the maximum length of passwords system value

iSeries Navigator	**Character-based interface**	**Description**
Maximum length	*maximum-number-of-characters*	Specify the maximum number of characters for a password.

Relationship to security policy

Within your security policy you should describe the password rules that are defined by the system values related passwords. For this system value, let users know the maximum number of characters that a valid

password contains. This system value works with the minimum length of password system value to provide a range for the length of passwords. Valid passwords must be within that length range. Remember that the length of passwords is dependant on the system value for password level. Although you may not want to disclose the password level to system users, you will want to document the length of passwords that the password level allows. For example, if you set your password level at 3, users would need to know that their passwords could be in a range of 1 to 128 characters.

Table 60. Quick Reference. Provides details for the maximum length of passwords system value.

iSeries Navigator name	**Maximum length**
Character-based interface name	QPWDMAXLEN
Authority	All object access (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Password Policy** and select **Properties**. 3. On the **Validation** page, you will find the options for password length. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QPWDMAXLEN`.
Changes take effect	Immediately
Default value	6
Recommended value	This value depends on what you selected for password level.
Lockable	Yes
Special considerations	NA

For more in-depth information about this security value, see Chapter 3, ″Security System Values″ in Security Reference.

Restrict duplicate passwords:

This system value controls whether the password must be different from previous passwords.

This system value controls whether the password must be different from previous passwords. This value sets a number of previous passwords that are checked for duplicate passwords. This value provides additional security by preventing users from specifying passwords used previously. It also prevents a user whose password has expired from changing it and then immediately changing it back to the old password.

See Quick reference table for an overview of the restrict duplicate passwords system value.

Table 61. Possible values for the restrict duplicate passwords system value

iSeries Navigator	**Character-based interface**	**Description**
Password re-use cycle	*number-of-password-values-checked*	Specify the number of passwords that are checked for duplicates.

Relationship to security policy

Within your security policy you should describe the password rules that are defined by the system values related passwords. For this system value, inform users that they cannot recycle passwords before this value has exceeded. Password recycling allows users to choose between three or four favorite passwords, however; this poses a security threat to your system. To minimize this threat, use this system value with the password expiration system value to prevent a password from being reused for at least 6 months. For example, if you selected 30 days for password expiration interval and selected 10 passwords for the password re-use cycle, then a typical user, who changes passwords when warned by the system, will not repeat a password for approximately 9 months.

Table 62. Quick Reference. Provides details for the restrict duplicate passwords system value.

iSeries Navigator name	**Password re-use cycle**
Character-based interface name	QPWDRQDDIF
Authority	All object access (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Password Policy** and select **Properties**. 3. On the **Validation** page, you will find the options for password re-use. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QPWDRQDDIF`.
Changes take effect	Immediately
Default value	After one password
Recommended value	After 10 passwords
Lockable	Yes
Special considerations	Select a value of 10 or more to prevent the use of repeated passwords. It is recommended to use a combination of the Password expiration value and the Password reuse cycle value to prevent a password from being reused for at least 6 month

For more in-depth information about this security value, see Chapter 3, ″Security System Values″ in Security Reference.

Restricted characters for passwords:

This system value limits the use of certain characters in a password.

The valid characters are A through Z, 0 through 9, and special characters number (#), dollar ($), at (@), and underscore (_). 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.

See Quick reference table for an overview of the restrict character for passwords system value.

Table 63. Possible values for the restrict character for passwords system value

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

Relationship to security policy

Within your security policy you should describe the password rules that are defined by the system values related passwords. For this system value, inform users which characters are restricted. This system value works with other system values that specify the composition of individual passwords.

Table 64. Quick Reference. Provides details for the restricted character for passwords system value.

iSeries Navigator name	Restricted characters
Character-based interface name	QPWDLMTCHR
Authority	All object access (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Password Policy** and select **Properties**. 3. On the **Validation** page, you will find the option for restricted character. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QPWDLMTCHR`.
Changes take effect	Immediately
Default value	None
Recommended value	A, E, I, O, and U. You may also want to restrict special characters (#, $, and @) for compatibility with other systems.
Lockable	Yes
Special considerations	This system value can only be used for password levels 0 or 1. If you change this value and your password level is either 2 or 3, then the system will ignore the restricted character setting.

For more in-depth information about this security value, see Chapter 3, ″Security System Values″ in Security Reference.

Restrict consecutive digits in passwords:

This 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.

See the following table for an overview of the restrict consecutive digits in passwords system value.

Table 65. Possible values for the restrict consecutive digits in passwords system value

iSeries Navigator	Character-based interface	Description
Deselected	0 (Yes)	Numeric characters are allowed next to each other in passwords.
Selected	1 (No)	Numeric characters are not allowed next to each other in passwords.

Relationship to security policy

Within your security policy you should describe the password rules that are defined by the system values related passwords. For this system value, inform users on whether or not password can contain adjacent numeric characters. This value provides additional security by preventing users from specifying passwords that are easy to guess, such as the same character repeated several times. This system value works with other system values that specify the composition of individual passwords.

Table 66. Quick Reference. Provides details for the restrict consecutive digits in passwords in passwords system value.

iSeries Navigator name	Restrict consecutive digits
Character-based interface name	QPWDLMTAJC
Authority	All object access (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Password Policy** and select **Properties**. 3. On the **Validation** page, you will find the option to restrict consecutive digits. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QPWDLMTAJC`.
Changes take effect	Immediately
Default value	Deselected
Recommended value	Selected
Lockable	Yes
Special considerations	NA

For more in-depth information about this security value, see Chapter 3, ″Security System Values″ in Security Reference.

Limit repeating characters in passwords:

This 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 is 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’.

See Quick reference table for an overview of the **Limit repeating characters in passwords** system value.

Table 67. Possible values for the limit repeating characters in passwords system value

iSeries Navigator	Character-based interface	Description
Characters may be used more than once	0	The same characters can be used more than once in a password.
Characters may not be used more than once	1	The same character cannot be used more than once in a password.
Characters may not be used consecutively	2	The same character cannot be used consecutively in a password.

Relationship to security policy

Within your security policy you should describe all password rules that are defined by the system values related passwords. For this system value, inform users on whether or not password rules allow repeated characters. This value provides additional security by preventing users from specifying passwords that are easy to guess, such as the same character repeated several times. This system value works with other system values that specify the composition of individual passwords.

Table 68. Quick Reference. Provides details for the **Limit repeating characters in passwords** system value.

iSeries Navigator name	Restrict repeating characters
Character-based interface name	QPWDLMTREP
Authority	All object access (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Password Policy** and select **Properties**. 3. On the **Validation** page, you will find the option to restrict repeating characters. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QPWDLMTREP`.
Changes take effect	Immediately
Default value	Characters may be used more than once
Recommended value	Characters cannot be repeated consecutively
Lockable	Yes
Special considerations	When the password level 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'.

For more in-depth information about this security value, see Chapter 3, ″Security System Values″ in Security Reference.

Require different characters in each position in the password:

This system value controls the position of each character in a new password.

Controlling the position of each character in a new password 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 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'.

See Quick reference table for an overview of the require different characters in each position system value.

Table 69. Possible values for the require different characters in each position in the password

iSeries Navigator	Character-based interface	Description
Deselected	0	The same characters can be used in a position corresponding to the same position in the previous password.
Selected	1	The same character cannot be used more than once in a password.

Relationship to security policy

Within your security policy, you should describe the password rules that are defined by the system values-related passwords. For this system value, inform users on whether they can repeat the same character in the same position as previous passwords This system value works with other system values that specify the composition of individual passwords.

Table 70. Quick Reference. Provides details for the require different characters in each position.

iSeries Navigator name	Require a new character in each position
Character-based interface name	QPWDPOSDIF
Authority	All object access (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Password Policy** and select **Properties**. 3. On the **Validation** page, you will find the option to require a new character in each position. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QPWDPOSDIF`.
Changes take effect	Immediately
Default value	Deselected
Recommended value	Selected
Lockable	Yes
Special considerations	When the password level is 2 or 3, the test for repeated characters is case sensitive. This means that a lowercase character is not the same as an uppercase character.

For more in-depth information about this security value, see Chapter 3, "Security System Values" in Security Reference.

Require a numeric character in passwords:

This system value determines whether users will require a numeric character in new passwords.

This value provides additional security by preventing users from using all alphabetic characters.

See Quick reference table for an overview of the require different characters in each position system value.

Table 71. Possible values for the require a number in passwords system value

iSeries Navigator	Character-based interface	Description
Deselected	0	Numeric characters are not required in new passwords.
Selected	1	One or more numeric characters are required in new passwords

Relationship to security policy

Within your security policy you should describe the password rules that are defined by the system values related passwords. For this system value, inform users on whether they can will need at least one numeric character in a new password. This value works with other system values that specify the composition of individual passwords.

Table 72. Quick Reference.. Provides details for the require a number in passwords system value.

iSeries Navigator name	Require at least one digit
Character-based interface name	QPWDRQDDGT
Authority	All object access (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.
How to access	**iSeries Navigator** 1. Expand **Security** → **Policies**. 2. Right click **Password Policy** and select **Properties**. 3. On the **Validation** page, you will find the option to require a numeric character in new passwords. **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QPWDRQDDGT`.
Changes take effect	Immediately
Default value	Deselected
Recommended value	Selected
Lockable	Yes
Special considerations	NA

For more in-depth information about this security value, see Chapter 3, ″Security System Values″ in Security Reference.

Store password information:

To support some network functions and communications requirements, iSeries servers provide a secure method for storing passwords that can be decrypted. Your system uses these passwords, for example, to establish a SLIP connection with another system.

Systems store these special passwords in a secure area that is not accessible to any user programs or interfaces. Only explicitly authorized system functions can set these passwords and retrieve them.

For example, when you use a stored password for dial-out SLIP connections, you set the password with the system command that creates the configuration profile (WRKTCPPTP). You must have *IOSYSCFG to use the command. A specially coded connection script retrieves the password and decrypts it during the dial-out procedure. The decrypted password is not visible to the user or in any job log.

As a security administrator, you need to decide whether you will allow passwords that can be decrypted to be stored on your system. You use the Retain Server Security Data (QRETSVRSEC) system value to specify this. The default is 0 (No). Therefore, your system will not store passwords that can be decrypted unless you explicitly set this system value.

If you have network or communications requirements for stored passwords, you should set appropriate policies and understand the policies and practices of your communications partners. For example, when you use SLIP to communicate with another iSeries server, both systems should consider setting up special user profiles for establishing the sessions. The special profiles should have limited authority on the system. This limits the impact to your system if a stored password is compromised on a partner system.

Password validation program:

This system value provides the ability for a user-written program to do additional validation on passwords.

The current and new passwords are passed to the validation program without encryption. The validation program could store passwords in a database file and compromise security on the system.

See the following table for an overview of the password validation program in each position system value.

Table 73. Possible values for the password validation program system value

Character-based interface	Description
*NONE	No validation program is used.
*REGFAC	The validation program name is retrieved from the registration facility.
program-specification	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 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 of the user changing the system value is used to search for the program. QSYS is the recommended library.
Note: There is no equivalent iSeries Navigator function for this system value.	

Relationship to security policy

A password validation program ensures that users are creating valid passwords that the system accepts; however, since new and old passwords are not encrypted when they are transferred to the validation program, they pose a security threat to your system. If the validation program stores passwords in a database file, an intruder could gain access and compromise security on the system. However if you decide that validating passwords is necessary to your enterprise, you should have any program that is designed inspected by your security officer and limit access to this program and any storage files it uses.

Table 74. Quick Reference. Provides details for the password validation program system value.

Character-based interface name	**QPWDVLDPGM**
Authority	All object access (*ALLOBJ) Security administrator (*SECADM) **Note:** The Security Officer (QSECOFR) user profile is shipped with these authorities.
How to access	**iSeries Navigator**: NA **Character-based interface** 1. In the character-based interface, type `WRKSYSVAL QPWDVLDPGM`.
Changes take effect	The next time a password is changed
Default value	*NONE
Recommended value	*NONE
Lockable	Yes
Special considerations	You must store a password validation program in the system auxiliary storage pool (ASP) or a basic user ASP.

For more information, see the section on using a password validation program in Chapter 3, "Security System Values" of the Security Reference manual.

Related information

Types of disk pools

Audit system values

This topic describes the auditing system values in detail.

You can specify the following system values to control security 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

To print the security system values, type: `WRKSYSVAL *SEC OUTPUT(*PRINT)`.

Related concepts

"Security audits" on page 17
This topic describes the purpose of security audits.

Related information

System value finder

Audit control:

The QAUDCTL system value allows you to control whether auditing is performed.

- Name in the character-based interface: **QAUDCTL**
- Name in the iSeries Navigator interface: **activate action auditing**, **activate object auditing**, and **do not audit objects in QTEMP**.
- **Description:** The QAUDCTL system value allows you to control whether auditing is performed. It functions like an on and off switch for the following:
  - The QAUDLVL and QAUDLVL2 system values.
  - The auditing defined for objects using the Change Object Auditing (CHGOBJAUD) and Change DLO Auditing (CHGDLOAUD) commands.
  - The auditing defined for users using the Change User Audit (CHGUSRAUD) command.

  You can specify more than one value for the QAUDCTL system value, unless you specify *NONE.
- **Recommended Values:** Allow the system to log events that are specified in the QAUDLVL system value (*AUDLVL); to log activity for objects that have object auditing defined (*OBJAUD); and to not audit objects in QTEMP (*NOQTEMP).
-

Table 75. Possible Values

Using the QAUDCTL system value	
***NONE**	No auditing of user actions and no auditing of objects is performed.
***OBJAUD**	Auditing is performed for objects that have been selected using the CHGOBJAUD, CHGDLOAUD, or CHGAUD commands.
***AUDLVL**	Auditing is performed for any functions selected on the QAUDLVL 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. You must specify this value with either *OBJAUD or *AUDLVL.

Note: This system value is a restricted value. For information on how to restrict changes to security system values and a list of the restricted system values, see "Chapter 3. Security System Values" in the *iSeries Security Reference*.

Audit level:

The QAUDLVL system value allows you to control which security-related events are logged to the security audit journal (QAUDJRN) for all system users.

- Name in the character-based interface: **QAUDLVL**
- Name in the iSeries Navigator interface: **activate action auditing**
- **Description:** The QAUDLVL system value allows you to control which security-related events are logged to the security audit journal (QAUDJRN) for all system users. This system value is controlled by the QAUDCTL system value. For the QAUDLVL system value to take effect, the QAUDCTL system value must include *AUDLVL. You can specify more than one value for the QAUDLVL system value, unless you specify *NONE.
- **Recommended Values:** The recommended values will log the following information on your system:
  - *AUTFAIL
    - All access failures (signon, authorization, job submission)
    - Incorrect password or user ID entered from a device
  - *PGMFAIL
    - Blocked instruction
    - Validation value failure
    - Domain violation

– *JOBDTA
  - Job start and stop data
  - Hold, release, stop, continue, change, disconnect, end, end abnormal, PSR-attached to prestart job entries

Table 76. Possible Values for the QAUDLVL System Value

Auditing 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.
*ATNEVT	Conditions that require further evaluation to determine the condition's security significance are audited.
*AUDLVL2	Both QAUDLVL and QAUDLVL2 system values will be used to determine the security actions to be audited.
*AUTFAIL	Authority failure events are logged.
*CREATE	Object create operations are logged.
*DELETE	Object delete operations are logged.
*JOBDTA	Actions that affect a job are logged.
*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: *NETBAS *NETCLU *NETFAIL *NETSCK
*NETFAIL	Network failures are audited.
*NETSCK	Socket tasks are audited.
*OBJMGT	Object move and rename operations are logged.
*OFCSRV	Changes to the system distribution directory and office mail actions are logged.
*OPTICAL	Use of Optical Volumes is logged.
*PGMADP	Obtaining authority from a program that adopts authority is logged.
*PGMFAIL	System integrity violations are logged.
*PRTDTA	Printing a spooled file, sending output directly to a printer, and sending output to a remote printer are logged.
*SAVRST	Restore operations are logged.
*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 76. Possible Values for the QAUDLVL System Value (continued)

Auditing value	Description
*SECURITY	Security-related functions are logged. *SECURITY is composed of several values to allow you to better customize your auditing: *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 system management functions is logged.

Note: This system value is a restricted value. For details on how to restrict changes to security system values and a list of the restricted system values, see "Chapter 3: Security System Values" in the *iSeries Security Reference*.

Audit level extension:

The QAUDLVL2 system value is required when more than sixteen auditing values are needed.

- Name in the character-based interface: **QAUDLVL2**.
- Name in the iSeries Navigator interface: **activate action auditing**.
- **Description:** 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.

Table 77. Possible Values for the QAUDLVL2 System Value

Auditing value	Description
*NONE	No auditing values are contained in this system value.
*ATNEVT	Conditions that require further evaluation to determine the condition's security significance are audited.
*AUTFAIL	Authority failure events are logged.
*CREATE	Object create operations are logged.
*DELETE	Object delete operations are logged.
*JOBDTA	Actions that affect a job are logged.
*NETBAS	Network base functions are audited.
*NETCLU	Cluster and cluster resource group operations are audited.

Table 77. Possible Values for the QAUDLVL2 System Value (continued)

Auditing value	Description
*NETCMN	Network and communication functions are audited. *NETCMN is composed of several values to allow you to better customize your auditing: *NETBAS *NETCLU *NETFAIL *NETSCK
*NETFAIL	Network failures are audited.
*NETSCK	Socket tasks are audited.
*OBJMGT	Object move and rename operations are logged.
*OFCSRV	Changes to the system distribution directory and office mail actions are logged.
*OPTICAL	Use of Optical Volumes is logged.
*PGMADP	Obtaining authority from a program that adopts authority is logged.
*PGMFAIL	System integrity violations are logged.
*PRTDTA	Printing a spooled file, sending output directly to a printer, and sending output to a remote printer are logged.
*SAVRST	Restore operations are logged.
*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.
*SECURITY	Security-related functions are logged. *SECURITY is composed of several values to allow you to better customize your auditing: *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 system management functions is logged.

Note: This system value is a restricted value. For details on how to restrict changes to security system values and a list of the restricted system values, see "Chapter 3: Security System Values" in the *iSeries Security Reference*.

Audit end action:

The QAUDENDACN system value allows you to set what action the system will take when audit records cannot be sent to the auditing journal because of errors that occur when the journal entry is sent.

- Name in the character-based interface: **QAUDENDACN**
- Name in the iSeries Navigator interface: **audit journal error action**
- **Description:** This system value allows you to set what action the system will take when audit records cannot be sent to the auditing journal because of errors that occur when the journal entry is sent.
- **Recommended Values:** For most installations, *NOTIFY is the recommended value. If your security policy requires that no processing be performed on the system without auditing, then you must select *PWRDWNSYS.

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

Table 78. Possible values

Using QAUDENDACN system value	
*NOTIFY	Message CPI2283 is sent to the QSYSOPR message queue and the QSYSMSG message queue (if it exists) every hour until auditing is successfully restarted. The system value QAUDCTL is set to *NONE to prevent the system from attempting to write additional audit journal entries. Processing on the system continues. If an IPL is performed before auditing is restarted, message CPI2284 is sent to the QSYSOPR and QSYSMSG message queues during the IPL.
*PWRDWNSYS	If the system is unable to write an audit journal entry, the system powers down immediately. The system unit displays system reference code (SRC) B900 3D10. When the system is powered on again, it is in a restricted state. This means the controlling subsystem is in a restricted state, no other subsystems are active, and sign-on is allowed only at the console. The QAUDCTL system value is set to *NONE. The user who signs on the console to complete the IPL must have *ALLOBJ and *AUDIT special authority.

Note: This system value is a restricted value. See Chapter 3: ″Security System Values″ in the Security Reference information for details on how to restrict changes to security system values and a complete list of the restricted system values. Also see the System Value Finder for more information.

Audit force level:

The QAUDFRCLVL system value allows you to set the number of journal entries written to the auditing journal before the journals entry data moves to auxiliary storage.

- Name in the character-based interface: **QAUDFRCLVL**
- Name in the iSeries Navigator interface: **maximum journal entries before writing to auxiliary storage**
- **Description:** This system value allows you to set the number of journal entries written to the auditing journal before the journals entry data moves to auxiliary storage. This system value controls the amount of auditing data that might be lost if the system ends abnormally.
- **Recommended Values:** *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 slow performance.

Table 79. Possible values for the QAUDFRCLVL System Value

QAUDFRCLVL System Value	Possible value
*SYS	The system determines when journal entries are written to auxiliary storage based on internal system performance.
number-of- records	Specify a number between 1 and 100 to determine how many audit entries can accumulate in memory before they are written to auxiliary storage. The smaller the number, the greater the impact on system performance.

Note: This system value is a restricted value. See Chapter 3: "Security System Values" in the Security Reference information for details on how to restrict changes to security system values and a complete list of the restricted system values. Also see the System Value Finder for more information.

Audit new objects:

The QCRTOBJAUD system value allows you to set the default auditing value used when new objects are created into a library.

- Name in the character-based interface: **QCRTOBJAUD**
- Name in the iSeries Navigator interface: **default auditing for default object**
- **Description:** The QCRTOBJAUD system value allows you to set the default auditing value used when new objects are created into a library. The QCRTOBJAUD system value is also the default object auditing value for new folderless documents.
- **Recommended Value:** The value you select depends upon the auditing requirements of your installation. You may also control the auditing value at the library level with the CRTOBJAUD parameter with the CRTLIB command and the CHGLIB command.

Table 80. Possible values for the QCRTOBJAUD System Value

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

Note: This system value is a restricted value. See Chapter 3: "Security System Values" in the iSeries Security reference information for details on how to restrict changes to security system values and a complete list of the restricted system values. Also see the System Value Finder for more information.

Security-related restore system values

Restoring programs to your system represents a security exposure.

A restored program may have been altered to perform functions that you do not intend, or the program may adopt the authority of a powerful user profile. These system values work together to determine the action the system takes regarding security-related objects. When preparing for a restore operation, you need to understand how the following security-related restore system values work together to restore objects securely.

- Verify object signatures during restore

- Force conversion on a restore
- Allow restore for security-sensitive objects
- Scan objects that are accessed after a restore operation

The verify object signature during restore system value controls the restore of digitally signed objects. Digital signatures provide enhanced integrity protection by ensuring that objects on the system have not been altered and come from a trusted source. This system value verify the signature on these objects by validating that the signer is trusted. If the object passes this system value without errors. The system then checks the value of force conversion on restore system value.

This second system value that the system checks determine whether to force the conversion objects during a restore operation. The force conversion on a restore system value allows you to specify whether or not to convert programs, service programs, SQL packages, and module objects during the restore. It can also prevent some objects from being restored. In addition to this system value, you can specify the Force object conversion (*FRCOBJCVN) parameter when you issue a restore command. Only objects that can get past the first two filters are processed by the third system value.

The allow restore of security-sensitive objects (QALWOBJRST) system value specifies whether or not objects with security-sensitive attributes can be restored.

Verify object on restore:

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 proper digital signature from a trusted software provider. This value applies to objects of types: *PGM, *SRVPGM, *SQLPKG, *CMD and *MODULE. It also applies to *STMF objects which contain Java programs.

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

If Digital Certificate Manager, (i5/OS option 34, is not installed on the system, all objects except those signed by a system trusted source are treated as unsigned when determining the effects of the QVFYOBJRST system value during a restore operation. A change to this system value takes effect immediately.

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.
- 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 i5/OS operating system.

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

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

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

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

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

Recommended Value: 3.

Force conversion on restore:

This system value allows you to specify whether or not to convert some object types during a restore. You can also use it to prevent some objects from being restored.

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

- Program (*PGM) v service program (*SRVPGM)
- SQL Package (*SQLPKG)
- Module (*MODULE)

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

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

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

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

Note: This system value is a restricted value. See Chapter 3: ⌂Security System Values⌂ in the iSeries Security Reference for details on how to restrict changes to security system values and a complete list of the restricted system values.

Allow restore for security-sensitive objects:

Three system values, Verify Object on Restore (QVFYOBJRST), Force Conversion on Restore (QFRCCVNRST), and Allow Object Restore (QALWOBJRST), act as a series of filters to determine whether a program is restored without change, whether it is re-created as it is restored, or whether it is not restored to the system.

The QVFYOBJRST system value determines whether objects are required to have digital signatures 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.

The QFRCCVNRST system value allows you to specify whether to convert the following object types during a restore:

- Program (*PGM)
- Service program (*SRVPGM)
- Module (*MODULE)
- SQL Package (*SQLPKG)

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

Before running a restore operation, you must plan what type of restore you want to perform. Then, configure your system values to the proper settings to meet your needs. Then, when a restore operation is performed, you will have the correct settings specified on your system. To plan how you want objects restored on the system, answer the following questions based on your company's needs:

- How cautious do you want to be about what is restored?
- What objects do you want to allow to be restored?

For more information on using these restore system values, see the following sections in Chapter 3 of the iSeries Security Reference:

- "Verify Object on Restore (WVFYOBJRST)"
- "Force Conversion on Restore (QFRCCVNRST)"
- "Allow Restoring of Security-Sensitive Objects (QALWOBJRST)"

Scan objects that are accessed after a restore operation:

The *NOPOSTRST value of the system value, QSCANFSCTL, impacts whether or not objects are scanned after a restore operation. Do you want to scan objects on the next access after the restore is complete? You need to consider what objects you are restoring and what kind of performance impact the scan will cause. Before determining whether or not to scan objects consider the following: Scanning may not be necessary if you are restoring your own objects which were saved with the option to scan objects and not save objects that failed the scan. Scanning may not be necessary if you are restoring objects that are coming from a trusted source.

System values selection worksheet

This topic introduces the System Values Selection worksheet.

Table 81. System Values Selection worksheet

General security system values			
Prepared by:		Date:	
System Value	**Recommended value**	**Your choice**	
System name			
Date separator (QDATSEP)			
Date format (QDATFMT)			
QSCANFS			
QSCANFSCTL			
Time separator (QTIMSEP)			
Device naming format for new devices (QDEVNAMING)	1 (system)		
System printer (QPRTDEV)			
Security level (QSECURITY)	40		
Allow security officers to sign on to any display station (QLMTSECOFR)	N		
Save job accounting information about completed printer output (QACGLVL)	N (*NONE)		

System Values Selection worksheet		Part 2 of 2
Additional instructions for Part 2 • Use the Work With System Value (WRKSYSVAL) command to enter Part 2.		
Security system values		
System value	**Recommended choice**	**Your choice**
Inactive job time-out interval (QINACTITV)	30 to 60	
Inactive job message queue (QINACTMSGQ)	*DSCJOB	
Limit device sessions (QLMTDEVSSN)	1 (YES)	
Action to take for failed sign-on attempts (QMAXSGNACN)	3 (Disable both)	
Maximum sign-on attempts allowed (QMAXSIGN)	3 to 5	
Password expiration interval (QPWDEXPITV)	30 to 60	
Maximum password length (QPWDMAXLEN)	8	
Minimum password length (QPWDMINLEN)	6	
Require different passwords (QPWDRQDDIF)	7 (6 unique passwords)	
Other system values		
System value	**Recommended choice**	**Your choice**
Disconnected job time-out interval (QDSCJOBITV)	300	
Note: You may want to set some other security-related system values. See Chapter 3 of the *Security Reference* (SC41-5302-04) for the complete list of security-related system values and the recommendations for them.		

Security considerations for internet browsers

Use this information to learn about common security threats from using Internet browsers.

Many PC users in your organization have browsers on their workstations. They might connect to the Internet or to your server. More information about the security considerations both for the PCs and for your server is available under Planning internet security in the System internet security topic for more info.

Risk: Workstation damage:

This topic describes security risks to workstations and provides recommendations for reducing these risks.

A Web page that your user visits might have an associated “program,” such as a Java applet, an Active-X control, or some other type of plug-in. This type of “program” when run on a PC has the potential to damage the information on the PC. As a security administrator, consider the following for protecting PCs in your organization:

- Understand the security options of the different browsers that your users have. For example, to prevent Java applets from damaging PC data, you can control the access that Java applets have outside the browser.

- Make recommendations to your users about their browser settings. You must educate users about the potential risks of improper settings.

Risk: Access to system directories through mapped drives:

This topic describes security risks to system directories and provides recommendations for reducing these risks.

Assume that a PC is connected to your server with an IBM iSeries Access for Windows session. The session set up mapped drives to link to the system's integrated file system. For example, the PC's G drive might map to the integrated file system of the SYSTEM1 server in the network.

Now assume that the same PC user has a browser and can access the Internet. The user requests a Web page that runs a mischievous "program" such as a Java applet or an Active-X control. Conceivably, the program could attempt to erase everything on the PC's G drive.

You have several protections against damage to mapped drives:
- Your most important protection is resource security on your server. The Java applet or Active-X control looks to the server like the user who established the PC session. You need to carefully manage what each PC user is authorized to do on your server.
- Advise your PC users to set their browsers to prevent attempts to access mapped drives. This works for Java applets but not for Active-X controls.
- Educate your users about the dangers of being connected to your server and the Internet in the same session. Also, make sure your PC users understand that drives remain mapped even when the iSeries Access session appears to be ended.

Risk: Trust signed applets:

This topic describes security risks from signed Java applets and provides recommendations for reducing these risks.

Your users might have followed your advice and set up their browsers to prevent applets from writing to any PC drives. However, your PC users need to be aware that a signed applet can override the setting for their browser.

A signed applet has an associated digital signature to establish its authenticity. When a user accesses a Web page that has a signed applet, the user sees a message. The message indicates the applet's signature, who signed it and when it was signed. When your user accepts the applet, the user grants the applet an override to the security settings for the browser. The signed applet can write to the PC's local drives, even though the default setting for the browser prevents it. The signed applet can also write to mapped drives on your server because they appear to the PC to be local drives.

For your own Java applets that come from your server, you might need to use signed applets. However, you should instruct your users in not to accept signed applets from unknown sources.

Plan LPAR security

Use this information to plan security for logical partitions (LPARs) on your servers.

Logical partitions allow you to distribute resources within a single server to make it function as if it were two or more independent servers. Each logical partition operates as an independent logical server. However, each partition shares a few physical system attributes such as the system serial number, system model, and processor feature code.

The security-related tasks that you perform on a partitioned system are the same as on a system without logical partitions. However, when you create logical partitions, you work with more than one independent system. Therefore you will have to perform the same tasks on each logical partition instead of just once on a system without logical partitions.

See ″Manage security for logical partitions″ under Systems Management.

Plan operations console security

Operations Console allows you to use your PC to access and control your system. It is important to include Operations Console in your overall security plan.

You can do any tasks that you could do from a traditional console from Operations Console. For example, user profiles that have *SERVICE or *ALLOBJ special authority are able to sign on to the Operations Console session, even if they are disabled.

Operations Console uses Service Tools User Profiles and passwords to enable the connection to the iSeries server. This makes it especially important to change your Service Tools User Profiles and passwords. Hackers are likely to be familiar with the default Service Tools User Profiles userids and passwords, and could use them to attempt a remote console session to your iSeries server. See “Change known passwords” on page 70 and “Avoid default passwords” on page 71 for tips on passwords.

Plan user security

Planning user security includes planning all areas where security affects the users on your system.

When you plan user security, it is essential that you describe the following areas:

User group security
: A user group is a group of users who need to use the same applications in the same way. Planning user group security involves determining the work groups who plan to use the system and the application needs of those groups.

Individual user security
: After you have determined what user groups you need, you can plan the individual user profiles that you need.

You might find the following planning forms helpful when planning user security:

- Use the physical security planning worksheet to describe any security issues that are related to the physical location of your system unit and attached devices.
- Use the user group ID worksheet to identify groups of users who have similar application needs.
- Use the user group description worksheet to describe characteristics of each user group.
- Use the system responsibilities worksheet to create a list of everyone that has access to your system that has a user class other than *USER.
- Create group profiles by filling out a user profile worksheet for each user group that is on your system, recording information about individual system users.

After you finish planning user security, you can start planning resource security.

Related concepts

“User security” on page 12
From a user’s point of view, security affects how they use and complete tasks on the system.

Plan user groups

This topic describes what to do to prepare for planning user groups.

The first step in the planning process, deciding your security strategy, is like setting company policy. Now you are ready to plan for groups of users, which is like deciding department policy.

What is a user group? A user group is exactly what its name implies: a group of people who need to use the same applications in the same way. Typically, a user group consists of people who work in the same department and have similar job responsibilities. You define a user group by creating a group profile.

What does a group profile do? A group profile serves two purposes on the system:

- **Security tool:** A group profile provides a simple way to organize who can use certain objects on your system (object authorities). You can define object authorities for an entire group rather than for each individual member of the group.
- **Customizing tool:** You can use a group profile 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 in the group profile and copy them to the individual user profiles.

Group profiles make it easier for you to maintain a simple, consistent scheme for both security and customizing.

What forms do you need?

- Complete a user group ID worksheet to identify the groups of users on your system that have similar application needs.
- Complete a user group description for each group that uses your system.

To complete these forms, you will need to perform the following tasks:

1. Identify user groups
2. Plan group profiles
3. Choose values that affect sign on
4. Choose values that limit what a user can do
5. Choose values that set up the user's environment

Identifying user groups

When you plan your user groups, you must first identify groups of users on your system. This allows you to plan accesses to resources that these groups need. Try using a simple method to identify your user groups. Think about the departments or work groups who plan to use the system. Look at the application diagram you drew earlier of your applications. See if a natural relationship exists between work groups and applications:

- Can you identify a primary application for each work group?
- Do you know which applications each group needs? Which applications they do not need?
- Do you know which group should own the information in each application library?

If you can answer ⌂Yes⌂ to those questions, then you can begin to plan your user groups. However, if you answered ⌂sometimes⌂ or ⌂maybe⌂, then you might find it helpful to use a systematic approach to identify your user groups.

Note: Making users a member of only one group profile simplifies your security management. However, some situations can benefit from having users belong to more than one group profile. Having users belong to more than one group profile is usually easier to manage than giving many private authorities to individual user profiles.

Decide what your user groups should be. Fill in the User Group Identification form, if you need it to help you decide. After you add your users to the User Group Identification form, you can plan a group profile.

Example: Identifying user groups

In this example, different groups need the Pricing and Contract application:

- The Sales and Marketing department sets prices and creating customer contracts. They own the pricing and contract information.
- The customer order department changes contract information indirectly. When they process orders, the quantities on the contract change. They need to change pricing and contract information.
- The order processing people need to look at the credit limit information to plan their work, but they are not allowed to change it. They need to view the credit limit file.

Table 82. Example: User Group Identification Form

User Group Identification Form					
		Access Needed for Applications			
User Name	**Department**	**APP:** A	**APP:** B	**APP:** C	**APP:** D
Ken H.	Order processing	O	C	C	C
Karen R.	Order processing	O	C	C	C
Kris T.	Accounting	V		V	O
Sandy J.	Accounting	V	C	V	O
Peter D.	Accounting	C		V	O
Ray W.	Warehouse	V	O	V	
Rose Q.	Warehouse	V	O	V	
Roger T.	Sales and marketing	C	C	O	C
Sharon J.	Management	C	C	C	C
Note: • Use a `V` (view) if someone only needs to look at the information in the application. • Use a `C` (change) if someone needs to make changes to the information. • Use an `O` (owner) if someone has primary responsibility for the information.					

Plan group profiles:

This topic describes the purpose of group profiles and how to design them. Use group profiles to define authorities for a group of users, rather than giving authority to each user individually.

A user can be a member of up to 16 group profiles. You can use a group profile as a pattern for creating individual user profiles.

Once you identify your user groups, you are ready to plan a profile for each group. Many of the decisions you make affect both security and customizing. For example, when you specify an initial menu, you might be restricting a user to only that menu. But you are also ensuring that the user sees the correct menu after signing on.

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

Security tool
: A group profile provides a method for organizing authorities on your system and sharing them among users. You can define object authorities or special authorities for group profiles rather than for each individual user profile. A user may be a member of up to 16 group profiles.

Customizing tool
: A group profile can be used as a pattern for creating individual user profiles. Most people who

are part of the same group have the same customizing needs, such as the initial menu and the default printer. You can define these things in the group profile and then copy the group profile to create individual user profiles.

A group profile is a useful tool when several users have similar security requirements. They are particularly useful when job requirements and group membership change. For example, if members of a department have responsibility for an application, a group profile can be set up for the department. As users join or leave the department, the group profile field in their user profiles can be changed. This is easier to manage than removing individual authorities from user profiles. You can create profiles specifically to be group profiles, or you can make an existing profile into a group profile. A group profile is simply a special type of user profile. It becomes a group profile when one of the following occurs:

- Another profile designates it as a group profile.
- You assign a group identification number (*gid*) to it.

For example:

1. Create a profile called GRPIC: `CRTUSRPRF GRPIC`
2. When the profile is created, it is an ordinary profile, not a group profile.
3. Designate GRPIC as the group profile for another group profile: `CHGUSRPRF USERA GRPPRF(GRPIC)`
4. The system now treats GRPIC as a group profile and assigns a *gid* to it.

Create a group profile plan

You create group profiles in the same way that you create individual profiles. The system recognizes a group profile when you add the first member to it. At that point, the system sets information in the profile indicating that it is a group profile. The system also generates a group identification number (*gid*) for the profile. You can also designate a profile as a group profile at the time that you create it by specifying a value in the GID parameter.

Perform the following steps to plan group profiles:

1. Prepare a user group description worksheet for each identified group.
2. Name groups consistently.
3. Use the naming conventions worksheet to document your group naming conventions.
4. Determine the application and library needs of each user group. Use the application descriptions and library description worksheets.
5. Define the job description for user groups.

Planning Primary Groups for Objects

Any object on the system can have a primary group. Primary group authority can provide a performance advantage if the primary group is the first group for most users of an object. Often, one group of users is responsible for some information on the system, such as customer information. That group needs more authority to the information than other system users. By using primary group authority, you can set up this type of authority scheme without affecting the performance of authority checking.

Planning Multiple Group Profiles

A user can be a member of up to 16 groups: the first group (GRPPRF parameter in the user profile) and 15 supplemental groups (SUPGRPPRF parameter in the user profile). By using group profiles, you can manage authority more efficiently and reduce the number of individual private authorities for objects. However, the misuse of group profiles can have a negative impact on the performance of authority checking.

Follow these suggestions when using multiple group profiles:

- Try to use multiple groups in combination with primary group authority and eliminate private authority to objects.
- Carefully plan the sequence in which group profiles are assigned to a user. The user's first group should relate to the user's primary assignment and the objects used most often. For example, assume a user called WAGNERB does inventory work regularly and does order entry work occasionally. The profile needed for inventory authority (DPTIC) should be WAGNERB's first group. The profile needed for order entry work (DPTOE) should be WAGNERB's first supplemental group. 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, be sure you understand how using multiple groups in combination with other authority techniques, such as authorization lists, may affect your system performance.

Prepare a user description worksheet

In this example, the "User group description worksheet" on page 103 includes the group profile name, the applications and libraries that the group uses.

Table 83. Example: User Group Description Worksheet

User Group Description Worksheet
Group profile name: `DPTWH` Description of the group: `Warehouse department`
Primary application for the group: `Inventory control` List other applications needed by the group: `None`
List each library that the group needs. Place an `X` in front of each library that should be in the initial library list for each group. • **X** `ITEMLIB` • **X** `ICPGMLIB`

Name group profiles

Because a group profile acts as a special type of user profile, you may want to identify group profiles on lists and displays. You need to assign them special names. To appear together on lists, your group profiles should begin with the same characters, such as GRP (for group) or DPT (for department). Use these guidelines when naming user groups:

- User group names can be up to 10 characters long.
- The name may include letters, numbers, and the special characters: pound (#), dollar ($), underline (_), and the at sign (@).
- The name cannot begin with a number.

Note: For each group profile, the system assigns a group identification number (*gid*). Normally, you can let the system generate a *gid*. If you use your system in a network, you may need to assign specific *gid*s to group profiles. Check with your network administrator to verify whether you need to assign IDs.

Determine the application and libraries a user group needs

If you have not already done so, add your user groups to the application diagram and libraries you drew earlier. This visual image will help you decide the resource and application needs of each group.

On Part 1 of the "User group description worksheet" on page 103, indicate the group's primary application, which is the application they use most often. List the other applications the group needs.

Look at your application description worksheet to see the libraries each group needs. Check with your programmer or application provider to find out the best method for providing access to these libraries. Most applications use one of these techniques:

- The application includes the libraries on a user's initial library list.
- The application runs a setup program which places the libraries in the user's library list.
- Libraries do not need to be in the library list. The application programs always specify the library.

The system uses a library list to find the files and programs you need when you run applications. The library list is a list of libraries the system searches for objects needed by the user. It has two parts:

1. System portion: Specified in the QSYSLIBL system value, the system portion is used for i5/OS libraries. The default for this system value does not need to be changed.
2. User portion: The QUSRLIBL system value provides the user portion of the library list. The user's job description specifies the initial library list, or commands after the user is signed on. If you have an initial library list, it overrides the QUSRLIBL system value. Application libraries should be included in the user portion of the library list.

Define the job description

When a user signs on the system, the user's job description defines many characteristics of the job, including how the job prints, how batch jobs are run, and the initial library list. Your system comes with a job description, called QDFTJOBD, which you can use when creating group profiles. However, QDFTJOBD specifies the QUSRLIBL system value as the initial library list. If you want different groups of users to have access to different libraries when signing on, you should create unique job descriptions for each group.

List each library needed by the group on the User Group Description Form. If the library should be included on the initial library list in the group's job description, mark each library name on the form.

Related concepts

"Group profiles" on page 8
Group profiles define authority for a group of users.

User group identification worksheet:

This topic describes the user group identification worksheet.

Table 84. User group identification worksheet

User Group Identification worksheet								
Prepared by:		Date:						
Instructions: • Learn about this worksheet in "Planning user groups". • This worksheet helps you identify groups of users who have similar application needs. 1. List your major applications across the top of the worksheet. 2. List your users in the left-hand column. 3. Mark needed applications for every user. • You do not need to enter the information on this worksheet into the system.								
		Access needed for applications						
User name	Department	APP:	APP:	APP:	APP:	APP:	APP:	APP:

Table 84. User group identification worksheet (continued)

User Group Identification worksheet								
Note: • If you have a *relaxed* security environment, use an **X** to mark which applications users need. • If you have a *strict* security environment, you may need to use **C** (change) and **V** (view) to specify *how* the applications are used.								

User group description worksheet:

This topic describes the user group description worksheet.

Table 85. User group description worksheet (part 1 of 2)

User Group Description worksheet	**Part 1 of 2**
Prepared by:	Date:
Instructions for Part 1 • Learn how to prepare this worksheet in "Planning user groups." • Learn how to enter this worksheet in "Setting up user security." • Prepare a separate worksheet for each group which will use the system. • Use the Create Job Description (CRTJOBD) command to create a job description for the group. The job description has the initial library list of the group.	
Group profile name: Description of the group:	
Primary application for the group: List other applications needed by the group:	
List each library the group needs. Mark **X** for each library that should be in the initial library list for the group:	
Note: Look at the application description worksheet for each application that is listed in the previous section to find out which libraries that application uses.	

Table 86. User Group Description worksheet (part 2 of 2)

User Group Description worksheet		**Part 2 of 2**
Additional instructions for Part 2 • The tables below list all of the fields that appear on the Create User Profile display. The fields are divided into two groups: those where you need to make a choice and those where IBM recommends the default value. • Use the Work with User Profiles display or the Create User Profile (CRTUSRPRF) command to enter the information from this part of the form into your system.		
Choose values for these fields in the group profile:		
Field Name	**Recommended Choice**	**Your Choice**
Group profile name (User)		

Table 86. User Group Description worksheet (part 2 of 2) (continued)

User Group Description worksheet		**Part 2 of 2**
Password	*NONE	
User class (type of user)	*USER	
Current library (default library)	*same as group profile name*	
Initial program to call (sign on program)		
Initial program library		
Initial menu (first menu)		
Initial menu library		
Limit capabilities (restrict command line use)	*YES	
Text (user description)		
Job description	*same as group profile name*	
Job description library		
Group profile name (user group)	*NONE	
Print device (default printer)		
Output queue	*DEV	
Note: These fields are in the order in which they appear on the Create User Profile display (using F4).		
Use the system-supplied values (defaults) for the fields below:		
Accounting code	Keyboard buffering	Public authority
Assistance level	Language ID	Set password to expire
Attention program	Limit device sessions	Sort sequence
Coded character set ID	Maximum storage	Special authority
Country or Region ID	Message queue	Special environment
Display signon information	Password expiration interval	Status
Document password	Priority limit	User options
Note: The fields in this list are arranged in alphabetical order.		

Plan user profiles

This topic describes the purpose of user profiles and how to design them.

A user profile 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.

Now that you have decided on your overall security strategy and have planned user groups, you are ready to plan individual user profiles.

Consider the following issues when planning user profiles:

- Naming considerations for user profiles
- Responsibilities assigned to individual users
- Values for each user

Complete these worksheets to plan user profiles:

- Individual user profile worksheet
- System responsibilities worksheet

Refer to these completed worksheets when planning for user profiles:

- "User group description worksheet" on page 103
- Naming Conventions worksheet
- Your application description worksheet

Naming user profiles

Your user profile name is how you are identified to the system. You enter your user profile name in the User ID field of the Sign On display. Any work you do and printer output you create is associated with your user profile name. 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 8 characters.
- A user profile name may include letters, numbers, and the special characters: pound (#), dollar ($), underline (_), and the at sign (@). It may not begin with a number or underline (_).
- The system does not distinguish between uppercase and lowercase letters in a user profile name. If you enter lowercase alphabetic characters, the system translates them to uppercase characters.
- The displays and lists you use to manage user profiles show them in alphabetical order by user profile name.
- All IBM-supplied profiles begin with the letter Q. To keep your profiles separate from IBM-supplied profiles, avoid assigning user profile names that begin with the character Q.

Remember: One technique for assigning user profile names is to use the first 7 characters of the last name followed by the first character of the first name. This method makes user profile names easy to remember. Also, your lists and displays are then sequenced alphabetically by last name.

Roles of the User Profile

The user profile has several roles on the system:

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

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

Example: Naming Convention Worksheet for User Profile

Table 87. Example: Naming Convention Worksheet: User Profiles

User Name	**User Profile Name**
Anderson, George	ANDSERSOG
Anderson, Roger	ANDERSOR
Jones, Sharon	JONESS
Type of Object	**Naming Convention**
User profiles	Use the first 7 characters of the user's last name, followed by the first character of the user's first name. Descriptions of the user profile will be last name, first name.

Describe how you plan to name user profiles on the Naming Conventions worksheet, then you can determine who should be responsible for system functions and choose values for each user.

For more information on user profiles, see "Using the Create User Profile Command" in the iSeries Security Reference.

Related concepts

"User profiles" on page 7
Every system user must have a user identity before they can sign on to and use a system. This user identity is called a user profile.

"Create profiles for users in the group" on page 186
This topic describes how to create profiles for individual users.

"Create profiles for users not in a group" on page 190
Copy the first individual user profile to create additional members in the group. Look at each individual profile carefully when you create it with the copy method.

System responsibilities worksheet:

This topic describes the system responsibilities worksheet.

Table 88. System responsibilities worksheet

System Responsibilities worksheet			
Prepared by:			Date:
Instructions: • Learn about this worksheet in Planning individual user profiles. • Use this worksheet to list everyone who has a user class other than *USER. • Transfer information from this worksheet to the *User Class* column of the user profile worksheet.			
Who is your primary security officer? Who is your backup security officer?			
Profile name	User name	Class	Comments

User profile worksheet:

This topic describes the individual user profile worksheet.

Table 89. Individual user profile worksheet

Individual User Profile worksheet	
Prepared by:	Date:
Instructions: • Learn how to prepare this worksheet in Planning individual user profiles. • Use this worksheet to record information about individual system users. Fill out one worksheet for each user group (group profile) that is on your system. • Use the blank columns at the right for any additional fields that you wish to specify for individual users. • Learn how to enter this worksheet in Set up user security.	
Group profile names:	
Owner of objects created:	Group authority to objects created:

Table 89. Individual user profile worksheet (continued)

Individual User Profile worksheet						
Group authority type:						
Make an entry for each member of the group:						
User profile	Text (description)	User class	Limit capability			

Plan resource security

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

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

Defining Who Can Access Information

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

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

You define who can use an object in several ways:

- Public Authority: The public consists of anyone who is authorized to sign on to your system. Public authority is defined for every object on the system, although the public authority for an object may be *EXCLUDE. Public authority to an object is used if no other specific authority is found for the object.
- Private Authority: You can define specific authority to use, or not use, an object. You can grant authority to an individual user profile or to a group profile. An object has private authority if any authority other than public authority, object ownership, or primary group authority is defined for the object.
- User Authority: Individual user profiles may be given authority to use objects on the system. This is one type of private authority.
- Group Authority: Group profiles may be given authority to use objects on the system. A member of the group gets the group's authority unless an authority is specifically defined for that user. Group authority is also considered private authority.
- Object Ownership: Every object on the system has an owner. The owner has *ALL authority to the object by default. However, the owner's authority to the object can be changed or removed. The owner's authority to the object is not considered private authority.
- Primary Group Authority: You can specify a primary group for an object and the authority the primary group has to the object. Primary group authority is stored with the object and may provide better

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

See Plan object authority for more object authority information.

Planning resource security

Now that you have completed the process for planning users on your system, you can plan the resource security which protects objects on the system. In Resource security you learn how to set up resource security on your system.

System values and user profiles control who has access to your system and prevent unauthorized users from signing on. Resource security controls the actions that authorized system users can perform after they have signed on successfully. Resource security supports the main goals of security on your system to protect:

- Confidentiality of information
- Accuracy of information to prevent unauthorized changes
- Availability of information to prevent accidental or deliberate damage

You may plan resource security differently, depending on whether your company develops applications or purchases them. For applications you develop, you should communicate the requirements for security of the information to the programmer during the application design process. When you purchase applications, you need to determine your security needs and match those needs with the way your provider has designed your applications. The techniques described here should help you in both cases.

This information provides a basic approach to planning resource security. It introduces the main techniques and shows how you can use them. The methods described here will not necessarily work for every company and every application. Consult your programmer or application provider as you plan resource security.

The following sections are provided to help you plan resource security: [list of active links to children]

- Resource security
- Understanding types of authority
- Planning security for application libraries
- Determining ownership of libraries and objects
- Grouping objects
- Protecting printer output
- Protecting workstations
- Implementing resource security
- Planning your application installation

The following planning forms are helpful when planning system level security:

- Complete an Application description worksheet for each application on your system.
- Reference Plan object authority to plan how you will establish ownership and public authority to your applications after you load them.
- Use the Authorization list worksheet to list the objects that the list and the groups and individuals who have access to the list secure.
- Use the Printer Output Queue and Workstation Security worksheet to list any workstation or output queue that requires special protection.

Determining your objectives for your resource security: To begin to plan resource security, you must first understand your objectives. The system provides flexible implementation of resource security. It gives you the power to protect critical resources exactly the way you want. But resource security also introduces additional overhead to your applications. For example, whenever an application needs an object, the system must check the user's authority to that object. You must balance your need for confidentiality against the cost of performance. As you make resource security decisions, weigh the value of security against its cost. To prevent resource security from degrading the performance of your applications, follow these guidelines:

- Keep your resource security scheme simple.
- Secure only those objects that you need to secure.
- Use resource security to supplement, not replace, the other tools for protecting information, such as:
  - Limiting users to specific menus and applications.
  - Preventing users from entering commands by limiting capabilities in user profiles.

Begin your resource security planning by defining your objectives. You can define your security objectives on either the Application Description form or the Library Description form. The form that you use depends on how your information is organized in libraries.

Planning security for workstations: After planning resource security for printers and printer output, you can begin planning workstation security. On your Physical Security Plan, you listed workstations that represent a security risk because of their location. Use this information to determine which workstations you need to restrict.

You can encourage the people who use these workstations to be particularly aware of security. They should sign off whenever they leave their workstations. You may want to record your decision about sign off procedures for vulnerable workstations in your security policy. You can also limit which functions can be performed at those workstations to minimize the risks.

The easiest method for limiting function at a workstation is to restrict it to user profiles with limited function. You may choose to prevent people with security officer or service authority from signing on at every workstation. If you use the QLMTSECOFR system value to do this, people with security officer authority can sign on only at specifically authorized workstations. Prepare the workstation portion of the Output Queue and Workstation Security form.

Summary of resource security recommendations: After you finish planning workstation security, you can review the following resource security recommendations. The system offers many options for protecting the information on your system. This gives you the flexibility to design the resource security plan that is best for your company. But this wealth of options can also be confusing. This information demonstrated a basic approach to planning resource security that uses these guidelines:

- Move from the general to the specific:
  - Plan security for libraries. Deal with individual objects only when necessary.
  - Plan public authority first, followed by group authority, and individual authority.
- Make the public authority for new objects in a library (CRTAUT) the same as the public authority you defined for the majority of existing objects in the library.
- Try not to give groups or individuals less authority than the public has. This diminishes performance, may lead to mistakes later, and makes auditing difficult. If you know that everyone has at least the same authority to an object that the public has, it makes planning and auditing security easier.
- Use authorization lists to group objects with the same security requirements. Authorization lists are simpler to manage than individual authorities and aid in recovery of security information.
- Create special user profiles as application owners. Set the owner password to *NONE.
- Avoid having applications owned by IBM-supplied profiles, such as QSECOFR or QPGMR.

- Use special output queues for confidential reports. Put the output queue in the same library as the confidential information.
- Limit the number of people who have security officer authority.
- Be careful when granting *ALL authority to objects or libraries. People with *ALL authority can accidentally delete things.

To ensure that you have planned successfully for setting up resource security, you should have gathered the following information:
- Fill in Part 1 and Part 2 of the Library description forms for all your application libraries.
- On your Individual user profile forms fill in the Owner of objects created and Group authority over objects created fields.
- On your Naming conventions form describe how you plan to name authorization lists.
- Prepare Authorization List forms.
- Add authorization list information to your Library description forms.
- Prepare an Output queue and workstation security form.

Now you are ready to plan your application installation.

End of Step 3 for planning a security strategy; provide link to next step: “Plan network security” on page 160

Related concepts

“Resource security” on page 14
You can use resource security on the system to control the actions of authorized users after successful authentication.

Plan library security

This topic describes how to plan security for the libraries on your system.

Many factors affect how you choose to group your application information into libraries and manage libraries. This topic addresses some of the security issues associated with library design. To access an object, you need authority to the object itself and to the library containing the object. You can restrict access to an object by restricting the object itself, the library containing the object, or both.

Planning Libraries

A library is like a directory used to locate the objects in the library. *USE authority to a library allows you to use the directory to find objects in the library. The authority for the object itself determines how you can use the object. *USE authority to a library is sufficient to perform most operations on the objects in the library.

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.

Library security is effective only if these rules are followed:
- Libraries contain objects with similar security requirements.
- Users are not allowed to add new objects to restricted libraries. Changes to programs in the libraries are controlled. That is, application libraries should have public authority of *USE or *EXCLUDE unless users need to create objects directly into the library.
- Library lists are controlled.

Describing Library Security

As an application designer, you need to provide information about a library for the security administrator. The security administrator uses this information to decide how to secure the library and its objects. Typical information needed is:

- Any application functions which add objects to the library.
- Whether any objects in the library are deleted during application processing.
- What profile owns the library and its objects.
- Whether the library should be included on library lists.

See the following 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.
```

Use library security to complement menu security

To access an object in a library, you must have authority both to the object and to the library. Most operations require either *EXECUTE authority or *USE authority to the library. Depending on your situation, you may be able to use library authority as a simple means for securing objects. For example, assume that for the Order-Entry menu example, everyone who has authority to the Order Entry menu can use all of the programs in the ORDERPGM library.

Rather than securing individual programs, you can set the public authority to the ORDERPGM library to *EXCLUDE. You can then grant *USE authority to the library to specific user profiles, which will allow them to use the programs in the library. This assumes that public authority to the programs is *USE or greater. Library authority can be a simple, efficient method for administering object authority. However, you must ensure that you are familiar with the contents of the libraries that you are securing so that you do not provide unintended access to objects.

Planning security for application libraries: After you have determined your objectives for your resource security, you can begin planning security for application libraries. Choose one of your application libraries to work with as you follow the process described here. If your system stores files and programs in separate libraries, choose a library that contains files. When you finish the topic, repeat these steps for your remaining application libraries.

Review the information that you gathered about your the applications and libraries:

- Application Description form
- Library Description form
- User Group Description form for any groups that need the library
- Your diagram of applications, libraries, and user groups

Think about which groups need the information in a library, why they need it, and what they need to do with it. Determine the contents of the application libraries, as they contain the important application files. They may also contain other objects, most of which are programming tools to make the application work properly, such as:

- Work files
- Data areas and messages queues
- Programs
- Message files
- Commands
- Output queues

Most of the objects, other than files and output queues, do not represent a security exposure. They usually contain small amounts of application data, often in a format that is not easily intelligible outside the programs. You can list names and descriptions of all the objects in a library by using the Display Library command. For example, to list contents of the CONTRACTS library: `DSPLIB LIB(CONTRACTS) OUTPUT(*PRINT)` Next you need to decide what public authority you want to have for application libraries and program libraries.

Deciding public authority to application libraries: For purposes of resource security, the public means anyone you authorize to sign on to your system. Public authority allows a user access to an object if you have not any other more specific access. In addition to deciding public authority for objects already in the library, you can specify the public authority for any new objects added to the library later. To do this, you use the Create Authority (CRTAUT) parameter. Usually, public authority to library objects and library create authority for new objects should be the same.

The Create Authority (QCRTAUT) system value determines the system-wide public authority for new objects. IBM ships the QCRTAUT system value with *CHANGE. Avoid changing QCRTAUT, because many system functions use it. If you specify *SYSVAL for the CRTAUT of an application library, it uses the QCRTAUT system value (*CHANGE).

Use public authority as much as possible, for both simplicity and good performance. To determine what public authority to a library should be, ask these questions:

- Should everyone in the company have access to most of the information in this library?
- What kind of access should people have to the majority of the information in this library?

Concentrate on decisions for the majority of the people and the majority of the information. Later, you will learn how to deal with the exceptions. Planning resource security is often a circular process. You may discover that you need to make changes to public authority after considering the requirements for specific objects. Try several combinations of public and private authority to both objects and libraries before you choose one that meets your security and performance needs.

Ensuring adequate authority: *CHANGE authority to objects and *USE authority to a library are adequate for most application functions. However, you need to ask your programmer or application provider some questions to determine if certain application functions require more authority:

- Are any files or other objects in the library deleted during processing? Are any files cleared? Are members added to any files? Deleting an object, clearing a file, or adding a file member requires *ALL authority to the object.
- Are any files or other objects in the library created during processing? Creating an object requires *CHANGE authority to the library.

Deciding public authority to program libraries: Often, application programs are kept in a separate library from files and other objects. You are not required to use separate libraries for applications, but many

programmers use this technique when they design applications. If your application has separate program libraries, you need to decide the public authority to those libraries.

You can use *USE authority to both the library and the programs in the library to run programs sufficiently, but program libraries may have other objects that require additional authority. Ask your programmer a few questions:

- Does the application use data areas or message queues to communicate between programs? Are they in the program library? *CHANGE authority to the object is required for handling data areas and message queues.
- Are any objects in the program library, such as data areas, deleted during processing? *ALL authority to an object is required to delete the object.
- Are any objects in the program library, such as data areas, created during processing? *CHANGE authority to the library is required to create any new objects in the library.

Fill in all of the resource security information on both parts of the Library Description form except the library owner and the authorization list column. You then can determine ownership of libraries and objects.

Note: A knowledgeable programmer who has access to a library may be able to retain access to objects in the library even after you have revoked authority to the library. If a library contains objects with high security requirements, restrict the objects and the library for complete protection.

Determining ownership of libraries and objects: After you plan security for application libraries, you can decide ownership of libraries and objects. Each object is assigned an owner when it is created. The owner of the object automatically has all authority to the object, which includes authorizing others to use the object, changing the object, and deleting it. The security officer can perform these functions for any object on the system.

The system uses the profile of the object owner to track who has authority to the object. The system completes this function internally. This may not affect the user profile directly. However, if you do not plan object ownership properly, some user profiles can become very large.

When the system saves an object, the system also saves the name of the owning profile with it. The system uses this information if it restores the object. If the owning profile for a restored object is not on the system, the system transfers ownership to an IBM-supplied profile called QDFTOWN.

Recommendations: The recommendations below apply in many, but not all, situations. After reviewing the recommendations, discuss your ideas for object ownership with your programmer or application provider. If you purchase applications, you may not be able to control what profile owns libraries and objects. The application may be designed to prevent changes of ownership.

- Avoid using an IBM-supplied profile, such as QSECOFR or QPGMR, as an application owner. These profiles own many objects in IBM-supplied libraries and are already very large.
- Normally, a group profile should not own an application. Every member in the group has the same authority as the group profile, unless you specifically assign lower authority. In effect, you would be giving every member of the group complete authority to the application.
- If you plan to delegate responsibility for control of applications to managers in various departments, those managers could be the owners of all the application objects. However, the manager of an application might change responsibilities. If that is the case, then you would transfer ownership of all the application objects to a new manager.
- Many people use the technique of creating a special owner profile for each application with the password set to *NONE. The owning profile is used by the system to manage authorities for the application. The security officer, or someone with that authority performs the actual management of the application or it is delegated to managers with *ALL authority to particular applications.

Decide what profiles should own your applications. Enter the owner profile information on each Library Description form. You can now decide ownership and access for your user libraries.

Deciding ownership and access for user libraries: If your system has the IBM Query for iSeries licensed program or another decision support program, your users need a library for storing the query programs they create. Normally, this library is the current library in the user profile. If a user belongs to a group, you use a field in the user profile to specify whether the user or the group owns any objects created by the user.

If the user owns the objects, you can specify what authority the group members have to use the objects. You can also specify whether the group's authority is primary group authority or private authority. Primary group authority may provide better system performance. The Group authority to objects created field is not used if the owner of objects created is the group. Group members automatically have *ALL authority to any objects created.

Decide who should own and have access to user libraries. Enter your choices in the Owner of objects created and Group authority over objects fields on the Individual User Profile form. Now you are ready to begin grouping objects.

Grouping objects

After you have determined ownership of libraries and objects, you can begin grouping objects on the system. To simplify managing authorities, use an authorization list to group objects with the same requirements. You can then give the public, group profiles, and user profiles authority to the authorization list rather than to the individual objects on the list. The system treats every object that you secure by an authorization list the same, but you can give different users different authorities to the entire list.

An authorization list makes it easier to reestablish authorities when you restore objects. If you secure objects with an authorization list, the restore process automatically links the objects to the list. You can give a group or user the authority to manage an authorization list (*AUTLMGT). Authorization list management allows the user to add and remove other users from the list and to change the authorities for those users.

Recommendations:

- Use authorization lists for objects that require security protection and that have similar security requirements. Using authorization lists encourages you to think about categories of authority rather than individual authorities. Authorization lists also make it easier to restore objects and to audit the authorities on your system.
- Avoid complicated schemes that combine authorization lists, group authority, and individual authority. Choose the method that best suits the requirement, rather than using all of the methods at the same time.

You will also need to add the naming convention for authorization lists to your Naming conventions form. Once you have prepared an AuthorizationlList form, go back and add that information to your Library description form. Your programmer or application provider might have already created authorization lists. Be sure to check with them.

Library Security

Most objects on the system reside in libraries. To access an object, you need authority both to the object itself and the library in which the object resides. For most operations, including deleting an object, *USE authority to the object library is sufficient (in addition to the authority required for the object). Creating a new object requires *ADD authority to the object library. Appendix D shows what authority is required by CL commands for objects and the object libraries.

Using library security is one technique for protecting information while maintaining a simple security scheme. For example, to secure confidential information for a set of applications, you could do the following:

- Use a library to store all confidential files for a particular group of applications.
- Ensure that public authority is sufficient for all objects (in the library) that are used by applications (*USE or *CHANGE).
- Restrict public authority to the library itself (*EXCLUDE).
- Give selected groups or individuals authority to the library, using *USE or *ADD if the applications require it.

Although library security is a simple, effective method for protecting information, it may not be adequate for data with high security requirements. Highly sensitive objects should be secured individually or with an authorization list, rather than relying on library security.

Library Security and Library Lists

When a library is added to a user's library list, the authority the user has to the library is stored with the library list information. The user's authority to the library remains for the entire job, even if the user's authority to the library is revoked while the job is active. When access is requested to an object and *LIBL is specified for the object, the library list information is used to check authority for the library. If a qualified name is specified, the authority for the library is specifically checked, even if the library is included in the user's library list.

Note: If a user is running under adopted authority when a library is added to the library list, the user remains authorized to the library even when the user is no longer running under adopted authority. This represents a potential security exposure. Any entries added to a user's library list by a program running under adopted authority should be removed before the adopted authority program ends.

In addition, applications that use library lists rather than qualified library names have a potential security exposure. A user who is authorized to the commands to work with library lists could potentially run a different version of a program.

Determine library owner:

When you plan your application installation, you must first decide the user profiles and installation values for each application.

Determining user profiles and installation values for applications: Before you install an application that was created on another system, you may need to create one or more user profiles. The user profile that owns the application libraries and objects should exist on your system before you load the libraries on your system. Record the profiles you need to create for each library and what parameters the profiles need on the Application installation form.

To determine the installation values necessary, ask your programmer or application provider the following questions and record their answers on the Application installation form:

- What profile owns the application library?
- What profile owns the objects in the library?
- What is the public authority to the library (AUT)?
- What is the public authority for new objects (CRTAUT)?
- What is the public authority for objects in the library?
- What programs, if any, adopt the authority of the owner?

Find out whether your programmers or application provider have created any authorization lists for the application. Prepare an Authorization list form for each created authorization list or ask your programmer for information about the list. You can determine whether you should change any installation values.

Changing installation values for applications: Compare the information from the Application installation form with your resource security plan for the library on the Library description form. If they are different, you need to decide what changes to make after the application is installed.

Changing application ownership: If your programmer or application provider has created a special profile to own the application libraries and objects, consider using that profile, even if it does not match your naming conventions.

Transferring ownership of objects can take a long time and should be avoided. If one of the IBM-supplied group profiles, such as QSECOFR or QPGMR, owns the application, you should transfer ownership to another profile after you install the application. Sometimes programmers design applications to prevent changes in object ownership. Try to work within the restrictions and still meet your own requirements for managing security. However, if an IBM-supplied profile, such as QSECOFR, owns the application, you and your programmer or application provider need to develop a plan to change ownership. Ideally, you should change ownership before you install the application.

Changing public authority: When you save objects, you also save their public authority with them. When you restore an application library to your system, the library and all its objects will have the same public authorities they had when they were saved. This is true even if you saved the library on another system. The CRTAUT value for a library (public authority for new objects) does not affect objects that are restored. They are restored with their saved public authority, regardless of the CRTAUT for the library.

You should change the public authority of libraries and objects to match your plan on the Library description form. To ensure that you have planned your application installation completely, you should:

- Finish filling out your initial Application installation form. Then go back and prepare forms for each additional application.
- Review all your forms and make sure they are complete. Make copies of your forms and keep them in a secure location until you have installed your system and your licensed programs.

After you finished these planning tasks, you are ready to set up your user security.

Library description worksheet:

After you have described your naming conventions, you should describe the libraries on your system. Libraries identify and organize objects on your system.

Placing similar files together in one library allows users easy access to critical applications and files. You can also customize your users' authorities, so that they can access some libraries, but not others. Describe all libraries that are on your system for each application. You may need to prepare more than one Library description form. Fill out only the descriptive information about the library. When you plan resource security for the library you will fill out the rest of the Library description form. You will need to add information about authorities to the libraries later. See Planning security for application libraries for details on completing the remainder on the Library description form. Before you continue, be sure to complete the library and file parts of the Naming conventions worksheet, and the descriptive information on the Library description worksheet for each application library.

Table 90. Library Description worksheet

Library description worksheet	
Prepared by:	Date:

Table 90. Library Description worksheet (continued)

Library description worksheet	
Instructions: • Learn about this worksheet in Determine library ownership. • Use this worksheet to describe your main libraries and define resource security requirements for them. • Fill out one worksheet for each major application library on your system.	
Library name:	Descriptive name (text):
Briefly describe the function of this library:	
Define the security objectives for the library, such as whether any information is confidential:	
Public authority to the library:	
Public authority to objects in the library:	
Public authority for new objects (CRTAUT):	
Library owner:	

Naming conventions worksheet:

When you know how the system names objects, you can plan and monitor security, solve problems, and plan backup and recovery.

Most applications have rules for assigning names to objects, such as libraries, files, and programs. If your applications come from different sources, they probably each have their own unique naming system. Be sure to record all the naming conventions of applications and objects on the Naming conventions worksheet. List the rules your applications use for naming libraries and files. You may want to use the blank lines for other naming conventions, such as programs and menus. If your applications come from different sources, they probably each have unique naming conventions. Describe the naming conventions for each application. You may need to prepare more than one Naming conventions worksheet.

Table 91. Naming conventions worksheet

Naming conventions worksheet		
Prepared by:		Date:
Instructions • You do not need to enter information from this worksheet directly into the system. • Use this worksheet to describe how you will assign names to the objects on your system. Give examples of each one.		
Type of object	**Naming convention**	**Example**
Group profiles		
User profiles		
Authorization lists		
Libraries		
Files		
Calendars		
Devices		
Tapes		

Plan application security

This topic provides on overview for creating an application security plan for your company.

To plan the right security for your applications, you need to know:

- What information do you plan to store on the system?
- Who needs access to that information?
- What kind of access do people need? Do they need to change information or only view it?

As you go through these application planning topics, you answer the first question about what information you plan to store on your system. In subsequent topics, you decide who needs that information and what kind of access people need. You do not enter the application planning information into the system; however, you will need it when you set up users and resource security.

What is an application?

In the first planning step for application security, you need to describe the applications you plan to run on your system. An application is a group of functions that logically belong together. Usually, two different types of applications can run on your server:

- Business applications: Applications you buy or develop to perform specific business functions, such as order processing or inventory management.
- Special applications: Applications you provide that are used throughout your company to perform a variety of activities that are not specific to a business process.

What forms do you need?

- Application description form
- Library description form
- Naming conventions form

Describing your applications

At this point, you need to gather some general information about each of your business applications. Add information about your application to the appropriate fields on the Application Description form as described below. Later you can use this information to help you plan user groups and application security:

Application name and abbreviation
: Give the application a short name and an abbreviation that you can use as shorthand on forms and for naming objects that the application uses.

Descriptive information
: Briefly describe what the application does.

Primary menu and library
: Identify which menu is the primary menu for accessing the application. Indicate the library in which the menu is. Usually the primary menu leads to other menus with specific application functions. Users like to see the primary menu for their main application immediately after signing on the system.

Initial program and library
: Sometimes applications run an initial program that sets up background information for the user or does security checking. If an application has an initial program or setup program, list it on the form.

Application libraries
: Each application usually has a main library for its files. Include all libraries that the application uses, including program libraries and libraries that other applications own. For example, the JKL

Toy Company's customer order application uses the inventory library to get item balances and descriptions. You can use the relationship between libraries and applications to determine who needs access to each library.

Finding information about your applications

If you do not already know the information you need about your applications, you may need to contact your programmer or application provider. Here are some methods for gathering the information yourself, if you do not have access to this information about an application that runs on your system:

- Users of the application can probably tell you the name of the primary menu and library, or you can watch them sign on the system.
- If users see the application immediately after signing on, look at the Initial program field in their user profiles. This field contains the initial program to the application. You can use the DSPUSRPRF command to view the initial program.
- You can list the names and descriptions of all the libraries on your system. Use the DSPOBJD *ALL *LIB. This displays all libraries on your system.
- You can observe active jobs while users are running the application. Use the Work with Active Jobs (WRKACTJOB) command with intermediate assistance level to get detailed information about interactive jobs. Display jobs and look at both library lists and their object locks to find out which libraries are being used.
- You can display batch jobs in an application using the Work with User Jobs (WRKUSRJOB) command.

To ensure that you gather all the information you need to plan your application security, you should complete these tasks before continuing:

- Complete an Application description form for each of your business applications. Fill out the entire form, except the security requirements section. You will use that section to plan resource security for the application as described in the topic Resource security.
- Prepare an Application description form for each special application, if applicable. Using the form helps you determine how to provide access to the application.

Note: Preparing Application description forms for special applications from IBM, such as IBM Query for iSeries is optional. Access to the libraries used by these applications does not require any special planning. However, you may find it useful to gather the information and prepare the forms.

Drawing an application diagram

As you prepare your Application description and Library description forms, you may find it useful to draw a diagram showing the relationship between applications and libraries. A diagram will help you to plan both user groups and resource security.

Collecting some information about your applications and libraries now will help you with many security decisions you need to make. Look at this as a chance to become more knowledgeable about your system and applications. To ensure that you have gathered the application information that you need, you should:

- Complete an Application description form for each business application on your system.
- Prepare an Application description form for each special application on your system.
- Fill in the library and file sections of the Naming conventions form.
- Prepare a Library description form for each application library.
- Draw a diagram of the relationship between your applications and libraries.

When you have completed these forms, you can begin planning your overall security strategy.

Planning Applications to Prevent Large Profiles

Because of the potential impacts to performance and security, IBM strongly recommends the following to avoid profiles from becoming too full:

- Do not have one profile own everything on your system.

  Create special user profiles to own applications. Owner profiles that are specific to an application make it easier to recover applications and to move applications between systems. Also, information about private authorities is spread among several profiles, which improves performance. By using several owner profiles, you can prevent a profile from becoming too large because of too many objects. Owner profiles also allow you to adopt the authority of the owner profile rather than a more powerful profile that provides unnecessary authority.
- Avoid having applications owned by IBM-supplied user profiles, such as QSECOFR or QPGMR.

  These profiles own a large number of IBM-supplied objects and can become difficult to manage. Having applications owned by IBM-supplied user profiles can also cause security problems when moving applications from one system to another. Applications owned by IBM-supplied user profiles can also impact performance for commands, such as CHKOBJITG and WRKOBJOWN.
- Use authorization lists to secure objects.

  If you are granting private authorities to many objects for several users, you should consider using an authorization list to secure the objects. Authorization lists will cause one private authority entry for the authorization list in the user's profile rather than one private authority entry for each object. In the object owner's profile, authorization lists cause an authorized object entry for every user granted authority to the authorization list rather than an authorized object entry for every object multiplied by the number of users that are granted the private authority.

Plan object authority:

This information is helpful when planning object authority.

Your challenge as security administrator is to protect your organization's information assets without frustrating the users on your system. You need to make sure that users have enough authority to do their jobs without giving them the authority to browse throughout the system and to make unauthorized changes.

The i5/OS operating system provides integrated object security. Users must use the interfaces that the system provides to access objects. For example, if you want to access a database file, you must use commands or programs that are intended for accessing database files. You cannot use a command that is intended for accessing a message queue or a job log.

Whenever you use a system interface to access an object, the system verifies that you have the authority to the object that is required by that interface. Object authority is a powerful and flexible tool for protecting the assets on your system. Your challenge as a security administrator is to set up an effective object security scheme that you can manage and maintain.

Object authority enforcement

Whenever you try to access an object, the operating system checks your authority to that object. However, if the security level on your system (QSECURITY system value) is set to 10 or 20, every user automatically has authority to access every object because every user profile has *ALLOBJ special authority.

Object authority tip: If you are not sure whether you are using object security, check the QSECURITY (security level) system value. If QSECURITY is 10 or 20, you are not using object security. You must plan and prepare before you change to security level 30 or higher. Otherwise, your users may not be able to access the information that they need.

Object authority to system commands and programs

Following are several suggestions when you restrict authority to IBM-supplied objects:

- When you have more than one national language on your system, your system has more than one system (QSYS) library. Your system has a QSYSxxxx library for each national language on your system. If you are using object authority to control access to system commands, remember to secure the command in the QSYS library and in every QSYSxxx library on your system.
- The System/38™ library sometimes provides a command with function that is equivalent to the commands that you want to restrict. Be sure you restrict the equivalent command in the QSYS38 library.
- If you have the System/36™ environment, you may need to restrict additional programs. For example, the QY2FTML program provides System/36 file transfer.

Grouping objects

After you have determined ownership of libraries and objects, you can begin grouping objects on the system. To simplify managing authorities, use an authorization list to group objects with the same requirements. You can then give the public, group profiles, and user profiles authority to the authorization list rather than to the individual objects on the list. The system treats every object that you secure by an authorization list the same, but you can give different users different authorities to the entire list.

An authorization list makes it easier to reestablish authorities when you restore objects. If you secure objects with an authorization list, the restore process automatically links the objects to the list. You can give a group or user the authority to manage an authorization list (*AUTLMGT). Authorization list management allows the user to add and remove other users from the list and to change the authorities for those users.

Recommendations:

- Use authorization lists for objects that require security protection and that have similar security requirements. Using authorization lists encourages you to think about categories of authority rather than individual authorities. Authorization lists also make it easier to restore objects and to audit the authorities on your system.
- Avoid complicated schemes that combine authorization lists, group authority, and individual authority. Choose the method that best suits the requirement, rather than using all of the methods at the same time.

You will also need to add the naming convention for authorization lists to your Naming Conventions form. Once you have prepared an Authorization List form, go back and add that information to your Library Description form. Your programmer or application provider might have already created authorization lists. Be sure to check with them.

Defining how information can be accessed

Authority means the type of access allowed to an object. Different operations require different types of authority. Note: In some environments, the authority associated with an object is called the object's mode of access. Authority to an object is divided into three categories:

1. Object Authority defines what operations can be performed on the object as a whole.
2. Data Authority defines what operations can be performed on the contents of the object.
3. Field Authority defines what operations can be performed on the data fields.

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

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.
*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 a part of the parent key in a referential constraint

Description of Authority Types

Authority	Name	Functions allowed
*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 Authorization List Management for more information.		

Commonly used authorities

Certain sets of object and data authorities are commonly required to perform operations on objects. You can specify these system-defined sets of authority (*ALL, *CHANGE, *USE) instead of individually defining the authorities needed for an object. *EXCLUDE authority is different than having no authority. *EXCLUDE authority specifically denies access to the object. Having no authority means you use the public authority defined for the object. The following table shows the system-defined authorities available using the object authority commands and displays.

System-Defined Authority

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

The following table shows additional system-defined authorities that are available using the WRKAUT and CHGAUT commands.

System-Defined Authority

Authority	*RWX	*RW	*RX	*R	*WX	*W
Object Authorities						
*OBJOPR	X	X	X	X	X	X
*OBJMGT						
*OBJEXIST						

System-Defined Authority

Authority	*RWX	*RW	*RX	*R	*WX	*W
*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	

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

LAN Server Permissions

Authority	LAN server permissions
*EXCLUDE	None
Object authorities	
*OBJOPR	See note 1
*OBJMGT	Permission
*OBJEXIST	Create, Delete
*OBJALTER	Attribute
*OBJREF	No equivalent
Data authorities	
*READ	Read
*ADD	Create
*UPD	Write
*DLT	Delete
*EXECUTE	Execute
[1] Unless NONE is specified for a user in the access control list, the user is implicitly given *OBJOPR.	

Defining what information can be accessed

You can define resource security for individual objects on the system. You can also define security for groups of objects using either library security or an authorization list:

- **Library Security:**

  Most objects on the system reside in libraries. To access an object, you need authority both to the object itself and the library in which the object resides. For most operations, including deleting an object, *USE authority to the object library is sufficient (in addition to the authority required for the object). Creating a new object requires *ADD authority to the object library. Appendix D shows what authority is required by CL commands for objects and the object libraries.

  Using library security is one technique for protecting information while maintaining a simple security scheme. For example, to secure confidential information for a set of applications, you could do the following:

  – Use a library to store all confidential files for a particular group of applications.

- Ensure that public authority is sufficient for all objects (in the library) that are used by applications (*USE or *CHANGE).
- Restrict public authority to the library itself (*EXCLUDE).
- Give selected groups or individuals authority to the library (*USE, or *ADD if the applications require it).

Although library security is a simple, effective method for protecting information, it may not be adequate for data with high security requirements. Highly sensitive objects should be secured individually or with an authorization list, rather than relying on library security.

- **Library Security and Library Lists:**

  When a library is added to a user's library list, the authority the user has to the library is stored with the library list information. The user's authority to the library remains for the entire job, even if the user's authority to the library is revoked while the job is active.

  When access is requested to an object and *LIBL is specified for the object, the library list information is used to check authority for the library. If a qualified name is specified, the authority for the library is specifically checked, even if the library is included in the user's library list.

  If a user is running under adopted authority when a library is added to the library list, the user remains authorized to the library even when the user is no longer running under adopted authority. This represents a potential security exposure. Any entries added to a user's library list by a program running under adopted authority should be removed before the adopted authority program ends.In addition, applications that use library lists rather than qualified library names have a potential security exposure. A user who is authorized to the commands to work with library lists could potentially run a different version of a program. See Library Lists for more information.

- **Field Authorities:**

  Field authorities are now supported for database files. Authorities supported are Reference and Update. You can only administer these authorities through the SQL statements, GRANT and REVOKE. You can display these authorities through the Display Object Authority (DSPOBJAUT) and the Edit Object Authority (EDTOBJAUT) commands. You can only display the field authorities with the EDTOBJAUT command; you cannot edit them.

  Changes for field authorities include the following:

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

- **Security and the System/38™ Environment:**

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

Library QUSER38 is not shipped with the operating system. However, it can be created by anyone with enough authority to create a library. See the System/38 Environment Programming manual for more information about the System/38 Environment.

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

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

- **Directory Security:**

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

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

- **Authorization List Security:**

  You can group objects with similar security requirements using an authorization list. An authorization list, conceptually, contains a list of users and the authority that the users have to the objects secured by the list. Each user can have a different authority to the set of objects the list secures. When you give a user authority to the authorization list, the operating system actually grants a private authority for that user to the authorization list.

  You can also use an authorization list to define public authority for the objects on the list. If the public authority for an object is set to *AUTL, the object gets its public authority from its authorization list.

  The authorization list object is used as a management tool by the system. It actually contains a list of all objects which are secured by the authorization list. This information is used to build displays for viewing or editing the authorization list objects.

  You cannot use an authorization list to secure a user profile or another authorization list. Only one authorization list can be specified for an object.

  Only the owner of the object, a user with all object (*ALLOBJ) special authority, or a user with all (*ALL) authority to the object, can add or remove the authorization list for an object. Objects in the system library (QSYS) can be secured with an authorization list. However, the name of the authorization list that secures an object is stored with the object.

  In some cases, when you install a new release of the operating system, all the objects in the QSYS library are replaced. The association between the objects and your authorization list would be lost. See the topic Planning Authorization Lists for examples of how to use authorization lists.

  Authorization List Management: You can grant a special operational authority called Authorization List Management (*AUTLMGT) for authorization lists. Users with *AUTLMGT authority are allowed to add and remove the users' authority to the authorization list and change the authorities for those users. *AUTLMGT authority, by itself, does not give authority to secure new objects with the list or to remove objects from the list.

  A user with *AUTLMGT authority can give only the same or less authority to others. For example, assume USERA has *CHANGE and *AUTLMGT authority to authorization list CPLIST1. USERA can add USERB to CPLIST1 and give USERB *CHANGE authority or less. USERA cannot give USERB *ALL authority to CPLIST1, because USERA does not have *ALL authority.

  A user with *AUTLMGT authority can remove the authority for a user if the *AUTLMGT user has equal or greater authority to the list than the user profile name being removed. If USERC has *ALL authority to CPLIST1, then USERA cannot remove USERC from the list, because USERA has only *CHANGE and *AUTLMGT.

Using Authorization Lists to Secure IBM-Supplied Objects: You may choose to use an authorization list to secure IBM-supplied objects. For example, you may want to restrict the use of a group of commands to a few users. Objects in IBM-supplied libraries, other than the QUSRSYS and QGPL libraries, are replaced whenever you install a new release of the operating system. Therefore, the link between objects in IBM-supplied libraries and authorization lists is lost. Also, if an authorization list secures an object in QSYS and a complete system restore is required, the link between the objects in QSYS and the authorization list is lost. After you install a new release or restore your system, use the EDTOBJAUT or GRTOBJAUT command to re-establish the link between the IBM-supplied object and the authorization list.

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

Authority for new objects in a library

Every library has a parameter called CRTAUT (create authority). This parameter determines the default public authority for any new object that is created in that library. When you create an object, the AUT parameter on the create command determines the public authority for the object. If the AUT value on the create command is *LIBCRTAUT, which is the default, the public authority for the object is set to the CRTAUT value for the library.

For example, assume library CUSTLIB has a CRTAUT value of *USE. Both of the commands below create a data area called DTA1 with public authority *USE:

- Specifying the AUT parameter: `CRTDTAARA DTAARA(CUSTLIB/DTA1) + TYPE(*CHAR) AUT(*LIBCRTAUT)`
- Allowing the AUT parameter to default. *LIBCRTAUT is the default: `CRTDTAARA DTAARA(CUSTLIB/DTA1) + TYPE(*CHAR)`

The default CRTAUT value for a library is *SYSVAL. Any new objects created in the library using AUT(*LIBCRTAUT) have public authority set to the value of the QCRTAUT system value. The QCRTAUT system value is shipped as *CHANGE. For example, assume the ITEMLIB library has a CRTAUT value of *SYSVAL. This command creates the DTA2 data area with public authority of change: `CRTDTAARA DTAARA(ITEMLIB/DTA2) + TYPE(*CHAR) AUT(*LIBCRTAUT)`

Note: Several IBM-supplied libraries, including QSYS, have a CRTAUT value of *SYSVAL. If you change QCRTAUT to something other than *CHANGE, you may encounter problems. For example, devices are created in the QSYS library. The default when creating devices is AUT(*LIBCRTAUT).

The CRTAUT value for the QSYS library is *SYSVAL. If QCRTAUT is set to *USE or *EXCLUDE, public authority is not sufficient to allow sign-on at new devices.

The CRTAUT value for a library can also be set to an authorization list name. Any new object created in the library with AUT(*LIBCRTAUT) is secured by the authorization list. The public authority for the object is set to *AUTL.

The CRTAUT value of the library is not used during a move (MOVOBJ), create duplicate (CRTDUPOBJ), or restore of an object into the library. The public authority of the existing object is used.

If the REPLACE (*YES) parameter is used on the create command, then the authority of the existing object is used instead of the CRTAUT value of the library.

Create Authority (CRTAUT) Risks: If your applications use default authority for new objects created during application processing, you should control who has authority to change the library descriptions. Changing the CRTAUT authority for an application library could allow unauthorized access to new objects created in the library.

Authority for new objects in a directory

When you create a new object in a directory using the CRTDIR, MD or MKDIR commands, you specify the data authority and object authority that the public receives for the object. If you use the *INDIR option, the authority for the created directory is determined from the directory it is being created in. Otherwise, you can specify the specific desired authority.

Objects that adopt the owner's authority

Sometimes a user needs different authorities to an object or an application, depending on the situation. For example, a user may be allowed to change the information in a customer file when using application programs providing that function. However, the same user should be allowed to view, but not change, customer information when using a decision support tool, such as SQL.

A solution to this situation is 1) give the user *USE authority to customer information to allow querying the files and 2) use adopted authority in the customer maintenance programs to allow the user to change the files.

When an object uses the owner's authority, this is called adopted authority. Objects of type *PGM, *SRVPGM, *SQLPKG and Java programs can adopt authority. When you create a program, you specify a user profile (USRPRF) parameter on the CRTxxxPGM command. This parameter determines whether the program uses the authority of the owner of the program in addition to the authority of the user running the program.

The following applies to adopted authority:

- Adopted authority is added to any other authority found for the user.
- Adopted authority is checked only if the authority that the user, the user's group, or the public has to an object is not adequate for the requested operation.
- The special authorities (such as *ALLOBJ) in the owner's profile are used.
- If the owner profile is a member of a group profile, the group's authority is not used for adopted authority.
- Public authority is not used for adopted authority. For example, USER1 runs the program LSTCUST, which requires *USE authority to the CUSTMST file:
  - Public authority to the CUSTMST file is *USE.
  - USER1's authority is *EXCLUDE.
  - USER2 owns the LSTCUST program, which adopts owner authority.
  - USER2 does not own the CUSTMST file and has no private authority to it.
  - Although public authority is sufficient to give USER2 access to the CUSTMST file, USER1 does not get access. Owner authority, primary group authority, and private authority are used for adopted authority.
  - Only the authority is adopted. No other user profile attributes are adopted. For example, the limited capabilities attributes are not adopted.
- Adopted authority is active as long as the program using adopted authority remains in the program stack. For example, assume PGMA uses adopted authority:
  - If PGMA starts PGMB using the CALL command, these are the program stacks before and after the CALL command:

Adopted Authority and the CALL Command

Program stack before CALL command	Program stack after CALL command
QCMD . . . PGMA	QCMD . . . PGMA PGMB

Because PGMA remains in the program stack after PGMB is called, PGMB uses the adopted authority of PGMA. (The use adopted authority (USEADPAUT) parameter can override this.

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

Adopted Authority and the TFRCTL Command

Program stack before TFRCTL command	Program stack after TFRCTL command
QCMD . . . PGMA	QCMD . . . PGMB

PGMB does not use the adopted authority of PGMA, because PGMA is no longer in the program stack.

- If the program running under adopted authority is interrupted, the use of adopted authority is suspended. The following functions do not use adopted authority:
  – System request
  – Attention key (If a Transfer to Group Job (TFRGRPJOB) command is running, adopted authority is not passed to the group job)
  – Break-message-handling program
  – Debug functions

  Note: Adopted authority is immediately interrupted by the attention key or a group job request. The user must have authority to the attention-key-handling program or the group job initial program, or the attempt fails.

For example, USERA runs the program PGM1, which adopts the authority of USERB. PGM1 uses the SETATNPGM command and specifies PGM2. USERB has *USE authority to PGM2. USERA has *EXCLUDE authority to PGM2. The SETATNPGM function is successful because it is run using adopted authority. USERA receives an authority error when attempting to use the attention key because USERB's authority is no longer active.

- If a program that uses adopted authority submits a job, that submitted job does not have the adopted authority of the submitting program.
- When a trigger program or exit point program is called, adopted authority from previous programs in the call stack will not be used as a source of authority for the trigger program or exit point program.
- The program adopt function is not used when you use the Change Job (CHGJOB) command to change the output queue for a job. The user profile making the change must have authority to the new output queue.
- Any objects created, including spooled files, are owned by the user of the program or by the user's group profile, not by the owner of the program.
- Adopted authority can be specified on either the command that creates the program (CRTxxxPGM) or on the Change Program (CHGPGM) command.

- If a program is created using REPLACE(*YES) on the CRTxxxPGM command, the new copy of the program has the same USRPRF, USEADPAUT, and AUT values as the replaced program. The USRPRF and AUT parameters specified on the CRTxxxPGM parameter are ignored.
- Only the owner of the program can specify REPLACE(*YES) on the CRTxxxPGM command when USRPRF(*OWNER) is specified on the original program.
- Only a user who owns the program or has *ALLOBJ and *SECADM special authorities can change the value of the USRPRF parameter.
- You must be signed on as a user with *ALLOBJ and *SECADM special authorities to transfer ownership of an object that adopts authority.
- If someone other than the program's owner or a user with *ALLOBJ and *SECADM special authorities restores a program that adopts authority, all private and public authorities to the program are revoked to prevent a possible security exposure.

The Display Program (DSPPGM) and Display Service Program (DSPSRVPGM) commands show whether a program adopts authority (User profile prompt) and whether it uses adopted authority from previous programs in the program stack (Use adopted authority prompt). The Display Program Adopt (DSPPGMADP) command shows all the objects that adopt the authority of a specific user profile. The Print Adopting Objects (PRTADPOBJ) command provides a report with more information about objects that adopt authority. This command also provides an option to print a report for objects that changed since the last time the command was run.

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

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

Adopted Authority Risks and Recommendations: Allowing a program to run using adopted authority is an intentional release of control. You permit the user to have authority to objects, and possibly special authority, which the user would not normally have. Adopted authority provides an important tool for meeting diverse authority requirements, but it should be used with care:

- Adopt the minimum authority required to meet the application requirements. Adopting the authority of an application owner is preferable to adopting the authority of QSECOFR or a user with *ALLOBJ special authority.
- Carefully monitor the function provided by programs that adopt authority. Make sure these programs do not provide a means for the user to access objects outside the control of the program, such as command entry capability.
- Programs that adopt authority and call other programs must perform a library qualified call. Do not use the library list (*LIBL) on the call.
- Control which users are permitted to call programs that adopt authority. Use menu interfaces and library security to prevent these programs from being called without sufficient control.

Programs that ignore adopted authority

You may not want some programs to use the adopted authority of previous programs in the program stack. For example, if you use an initial menu program that adopts owner authority, you may not want some of the programs called from the menu program to use that authority.

The use adopted authority (USEADPAUT) parameter of a program determines whether the system uses the adopted authority of previous programs in the stack when checking authority for objects. When you create a program, the default is to use adopted authority from previous programs in the stack. If you do not want the program to use adopted authority, you can change the program with the Change Program (CHGPGM) command or Change Service Program (CHGSRVPGM) command to set the USEADPAUT parameter to *NO. If a program is created using REPLACE(*YES) on the CRTxxxPGM command, the new copy of the program has the same USRPRF, USEADPAUT, and AUT values as the replaced program.

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

Authority holders

An authority holder is a tool for keeping the authorities for a program-described database file that does not currently exist on the system. Its primary use is for System/36 environment applications, which often delete program-described files and create them again. An authority holder can be created for a file that already exists or for a file that does not exist, using the Create Authority Holder (CRTAUTHLR) command. The following applies to authority holders:

- Authority holders can only secure files in the system auxiliary storage pool (ASP) or a basic user ASP. They cannot secure files in an independent ASP.
- The authority holder is associated with a specific file and library. It has the same name as the file.
- Authority holders can be used only for program-described database files and logical files created in the S/36 environment.
- Once the authority holder is created, you add private authorities for it like a file. Use the commands to grant, revoke, and display object authorities, and specify object type *FILE. On the object authority displays, the authority holder is indistinguishable from the file itself. The displays do not indicate whether the file exists nor do they show that the file has an authority holder.
- If a file is associated with an authority holder, the authorities defined for the authority holder are used during authority checking. Any private authorities defined for the file are ignored.
- Use the Display Authority Holder (DSPAUTHLR) command to display or print all the authority holders on the system. You can also use it to create an output file (Outfile) for processing.
- If you create an authority holder for a file that exists:
  - The user creating the authority holder must have *ALL authority to the file.
  - The owner of the file becomes the owner of the authority holder regardless of the user creating the authority holder.
  - The public authority for the authority holder comes from the file. The public authority (AUT) parameter on the CRTAUTHLR command is ignored.
  - The existing file's authority is copied to the authority holder.
- If you create a file and an authority holder for that file already exists:
  - The user creating the file must have *ALL authority to the authority holder.
  - The owner of the authority holder becomes the owner of the file regardless of the user creating the file.
  - The public authority for the file comes from the authority holder. The public authority (AUT) parameter on the CRTPF or CRTLF command is ignored.
  - The authority holder is linked to the file. The authority specified for the authority holder is used to secure the file.
- If an authority holder is deleted, the authority information is transferred to the file itself.

- If a file is renamed and the new file name matches an existing authority holder, the authority and ownership of the file are changed to match the authority holder. The user renaming the file needs *ALL authority to the authority holder.
- If a file is moved to a different library and an authority holder exists for that file name and the target library, the authority and ownership of the file are changed to match the authority holder. The user moving the file must have *ALL authority to the authority holder.
- Ownership of the authority holder and the file always match. If you change the ownership of the file, ownership of the authority holder also changes.
- When a file is restored, if an authority holder exists for that file name and the library to which it is being restored, it is linked to the authority holder.
- Authority holders cannot be created for files in these libraries: QSYS, QRCL, QRECOVERY, QSPL, QTEMP, and QSPL0002 – QSPL0032.

Authority Holders and System/36 Migration: The System/36 Migration Aid creates an authority holder for every file that is migrated. It also creates an authority holder for entries in the System/36 resource security file if no corresponding file exists on the System/36. You need authority holders only for files that are deleted and re-created by your applications. Use the Delete Authority Holder (DLTAUTHLR) command to delete any authority holders that you do not need.

Authority Holder Risks: An authority holder provides the capability of defining authority for a file before that file exists. Under certain circumstances, this could allow an unauthorized user to gain access to information. If a user knew that an application would create, move, or rename a file, the user could create an authority holder for the new file. The user would thus gain access to the file. To limit this exposure, the CRTAUTHLR command is shipped with public authority *EXCLUDE. Only users with *ALLOBJ authority can use the command, unless you grant authority to others.

Working with authority

This information describes commonly-used methods for setting up, maintaining, and displaying authority information on your system. ″Security commands″ 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.

Authority displays

Four displays show object authorities:
- Display Object Authority display
- Edit Object Authority display
- Display Authority display
- Work with Authority display

Note:
- 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.
- The *ADOPTED authority indicates only the additional authority received from the program owner.

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)

Working with libraries

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

- Authority (AUT): The AUT parameter can be used to specify either of the following:
  - The public authority for the library
  - The authorization list that secures the library.

  The AUT parameter applies to the library itself, not to the objects in the library. If you specify an authorization list name, the public authority for the library is set to *AUTL. If you do not specify AUT when you create a library, *LIBCRTAUT is the default. The system uses the CRTAUT value from the QSYS library, which is shipped as *SYSVAL.
- Create Authority (CRTAUT): The CRTAUT parameter determines the default authority for any new objects that are created in the library. CRTAUT can be set to one of the system-defined authorities (*ALL, *CHANGE, *USE, or *EXCLUDE), to *SYSVAL (the QCRTAUT system value), or to the name of an authorization list.

  Note: You can change the CRTAUT value for a library using the Change Library (CHGLIB) command.

Creating objects

When you create a new object, you can either specify the authority (AUT) or use the default, *LIBCRTAUT.

Working with individual object authority

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

- *ALLOBJ authority or membership in a group profile that has *ALLOBJ special authority.

  Note: The group's authority is not used if you have private authority to the object.
- Ownership of the object. If a group profile owns the object, any member of the group can act as the object owner, unless the member has been given specific authority that does not meet the requirements for changing the object's authority.
- *OBJMGT authority to the object and any authorities being granted or revoked (except *EXCLUDE). Any user who is allowed to work with the object's authority can grant or revoke *EXCLUDE authority.

The easiest way to change authority for an individual object is with the Edit Object Authority display. This display can be called directly by using the Edit Object Authority (EDTOBJAUT) command or selected as an option from the Work with Objects by Owner (WRKOBJOWN) or WRKOBJ (Work with Objects) display. You can also use these commands to change object authority:

- Change Authority (CHGAUT)
- Work with Authority (WRKAUT)
- Grant Object Authority (GRTOBJAUT)
- Revoke Object Authority (RVKOBJAUT)

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

Specifying user-defined authority

The Object Authority column on the Edit Object Authority display allows you to specify any of the system-defined sets of authorities (*ALL, *CHANGE, *USE, *EXCLUDE). If you want to specify authority that is not a system-defined set, use F11 (Display detail).

Note: If the User options (USROPT) field in your user profile is set to *EXPERT, you always see this detailed version of the display without having to press F11. You can press F11 (Display data authorities) to view or change the data authorities.

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.

Removing a user's authority for an object is different from giving the user *EXCLUDE authority. *EXCLUDE authority means the user is specifically not allowed to use the object. Only *ALLOBJ special authority and adopted authority override *EXCLUDE authority. Removing a user's authority means the user has no specific authority to the object. The user can gain access through a group profile, an authorization list, public authority, *ALLOBJ special authority, or adopted authority.

You can remove a user's authority using the Edit Object Authority display. Type blanks in the Object Authority field for the user and press the Enter key. The user is removed from the display. You can also use the Revoke Object Authority (RVKOBJAUT) command. Either revoke the specific authority the user has or revoke *ALL authority for the user.

Note: The RVKOBJAUT command revokes only the authority you specify. For example, USERB has *ALL authority to FILEB in library LIBB. You revoke *CHANGE authority: `RVKOBJAUT OBJ(LIBB/FILEB) OBJTYPE(*FILE) + USER(*USERB) AUT(*CHANGE)`

Working with authority for multiple objects

The **Edit Object Authority** display allows you to interactively work with the authority for one object at a time. The Grant Object Authority (GRTOBJAUT) command allows you to make authority changes to more than one object at a time. You can use the GRTOBJAUT authority command interactively or in batch. You can also call it from a program. Following are examples of using the GRTOBJAUT command, showing the prompt display. When the command runs, you receive a message for each object indicating whether the change was made. Authority changes require an exclusive lock on the object and cannot be made when an object is in use. Print your job log for a record of changes attempted and made. To give all the objects in the TESTLIB library a public authority of *USE:

```
Grant Object Authority (GRTOBJAUT)

Type choices, press Enter.
Object . . . . . *ALL
Library . . . . . TESTLIB
Object type . . . . . *ALL
ASP device . . . . . *
Users . . . . . *PUBLIC
+ for more values
Authority . . . . . *USE
```

This example for the GRTOBJAUT command gives the authority you specify, but it does not remove any authority that is greater than you specified. If some objects in the TESTLIB library have public authority *CHANGE, the command just shown would not reduce their public authority to *USE. To make sure that all objects in TESTLIB have a public authority of *USE, use the GRTOBJAUT command with the REPLACE parameter: `GRTOBJAUT OBJ(TESTLIB/*ALL) OBJTYPE(*ALL) + USER(*PUBLIC) REPLACE(*YES)`

The REPLACE parameter indicates whether the authorities you specify replaces the existing authority for the user. The default value of REPLACE(*NO) gives the authority that you specify, but it does not remove any authority that is greater than the authority you specify, unless you are granting *EXCLUDE authority. These commands set public authority only for objects that currently exist in the library. To set the public authority for any new objects that are created later, use the CRTAUT parameter on the library description.

Working with object ownership

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

- The Change Object Owner (CHGOBJOWN) command
- The Work with Objects by Owner (WRKOBJOWN) command
- The Change Owner (CHGOWN) command

The Work with Objects by Owner display shows all the objects owned by a profile. You can assign individual objects to a new owner. You can also change ownership for more than one object at a time by using the NEWOWN (new owner) parameter at the bottom of the display. 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 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.

Resource security

Resource security on the system allows you to define who can use objects and how those objects can be used. The ability to access an object is called authority. When you set up object authority, you can need to be careful to give your users enough authority to do their work without giving them the authority to browse and change the system.

Object authority gives permissions to the user for a specific object and can specify what the user is allowed to do with the object. An object resource can be limited through specific detailed user authorities, such as adding records or changing records. System resources can be used to give the user access to specific system-defined subsets of authorities: *ALL, *CHANGE, *USE, and *EXCLUDE. Files, programs, libraries, and directories are the most common system objects that require resource security protection, but you can specify authority for any individual object on the system.

Understanding the types of authority: After you have determined your objectives for your resource security and recorded your decisions on the Library Description form, you can begin to plan types of authority. Resource security defines how users have access to objects on the system.

Authority means how someone is authorized to use an object. For example, you may have the authority to view information or to change information on the system. The system provides several different authority types. IBM groups these authority types into categories, called system-defined authorities, which meet the needs of most people. The table below lists the categories and tells how they apply to securing files and programs.

Note: Refer to the tables below when you plan authorities.

System-defined authorities

Authority name	Operations allowed for files	Operations not allowed for files	Operations allowed for programs	Operations not allowed for programs
*USE	View information in the file.	Change or delete any information in the file. Delete the file.	Run the program.	Change or delete the program

System-defined authorities

Authority name	Operations allowed for files	Operations not allowed for files	Operations allowed for programs	Operations not allowed for programs
*CHANGE	View, change and delete records in the file.	Delete or clear the entire file.	Change the description of the program.	Change or delete the program.
*ALL	Create and delete the file. Add, change and delete records in the file. Authorize others to use the file.	None	Create, change and delete the program. Authorize others to use the program.	Change the owner of the program, if the program adopts authority.
*EXCLUDE[1]	None	Any access to the file.	None	Any access to the program.
[1] *EXCLUDE overrides any authorities that you grant to the public or through a group profile.				

To design simple resource security, try to plan security for entire libraries. To do this, you need to understand how the system-defined authorities apply to libraries, which the table below shows:

System-defined authorities for libraries

Authority name	Operations allowed	Operations not allowed
*USE	• For objects in the library, any operation allowed by the authority to the specific object. • For the library, view descriptive information.	• Add new objects to the library. • Change the library description. • Delete the library.
*CHANGE	• For objects in the library, any operation allowed by the authority to the specific object. • Add new objects to the library. • Change the library description.	Delete the library.
*ALL	• Everything allowed with change. • Delete the library. • Authorize others to the library.	None.

You also need to understand how library and object authority work together. The table below gives examples of authorities that are required for both an object and the library:

How library authority and object authority work together

Object type	Operations	Object authority needed	Library authority needed
File	Change data	*CHANGE	*USE
File	Delete the file	*ALL	*USE
File	Create the file	*ALL	*CHANGE
Program	Run the program	*USE	*USE
Program	Change (recompile) the program	*ALL	*CHANGE
Program	Delete the program	*ALL	*USE

Directory authority is similar to library authority. You need authority to all the directories in the path name for an object in order to access the object.

Determine object ownership:

Every object on the system has an owner. The owner has *ALL authority to the object by default.

Object Ownership

Each object is assigned an owner when it is created. The owner is either the user who creates the object or the group profile if the member user profile has specified that the group profile should be the owner of the object. When the object is created, the owner is given all the object and data authorities to the object.

The owner of an object always has all the authority for the object unless any or all authority is removed specifically. As an object owner, you may choose to remove some specific authority as a precautionary measure. For example, if a file exists that contains critical information, you may remove your object existence authority to prevent yourself from accidentally deleting the file. However, as object owner, you can grant any object authority to yourself at any time.

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

When changing an object's owner, you have the option to keep or revoke the former owner's authority. A user with *ALLOBJ authority can transfer ownership, as can any user who has the following:

- Object existence authority for the object, except for an authorization list
- Ownership of the object, if the object is an authorization list
- Add authority for the new owner's user profile
- Delete authority for the present owner's user profile

You cannot delete a profile that owns objects. Ownership of objects must be transferred to a new owner or the objects must be deleted before the profile can be deleted. The Delete User Profile (DLTUSRPRF) command allows you to handle owned objects when you delete the profile.

Object ownership is used as a management tool by the system. The owner profile for an object contains a list of all users who have private authority to the object. This information is used to build displays for editing or viewing object authority.

Profiles that own many objects with many private authorities can become very large. The size of a profile that owns many objects affects performance when displaying and working with the authority to objects it owns, and when saving or restoring profiles. System operations can also be impacted. To prevent impacts to either performance or system operations, do not assign objects to only one owner profile for your entire system. Each application and the application objects should be owned by a separate profile. Also, IBM-supplied user profiles should not own user data or objects. The owner of an object also needs sufficient storage for the object.

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. There are several situations that cause ownership of an object to be assigned to the QDFTOWN profile:

- If an owning profile becomes damaged and is deleted, its objects no longer have an owner. Using the Reclaim Storage (RCLSTG) command assigns ownership of these objects to the default owner (QDFTOWN) user profile.
- If an object is restored and the owner profile does not exist.

- If a program that needs to be created again is restored, but the program creation is not successful.
- 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. The QDFTOWN user profile is intended for system use only. You should not design your security such that QDFTOWN normally owns object.

Changing application ownership

If your programmer or application provider has created a special profile to own the application libraries and objects, consider using that profile, even if it does not match your naming conventions. Transferring ownership of objects can take a long time and should be avoided. If one of the IBM-supplied group profiles, such as QSECOFR or QPGMR, owns the application, you should transfer ownership to another profile after you install the application. Sometimes programmers design applications to prevent changes in object ownership. Try to work within the restrictions and still meet your own requirements for managing security. However, if an IBM-supplied profile, such as QSECOFR, owns the application, you and your programmer or application provider need to develop a plan to change ownership. Ideally, you should change ownership before you install the application.

Changing public authority

When you save objects, you also save their public authority with them. When you restore an application library to your system, the library and all its objects will have the same public authorities they had when they were saved. This is true even if you saved the library on another system. The CRTAUT value for a library does not affect objects that are restored. They are restored with their saved public authority, regardless of the CRTAUT for the library. You should change the public authority of libraries and objects to match your plan on the Library description form.

Group ownership of objects:

This topic discusses security differences when an object is owned by a group, not an individual.

Group Ownership of Objects: When an object is created, the system looks at the profile of the user creating the object to determine object ownership. If the user is a member of a group profile, the OWNER field in the user profile specifies whether the user or the group should own the new object.

If the group owns the object, OWNER is *GRPPRF, the user creating the object is not automatically given any specific authority to the object. The user gets authority to the object through the group. If the user owns the object, OWNER is *USRPRF, the group's authority to the object is determined by the GRPAUT field in the user profile.

The group authority type, GRPAUTTYP field in the user profile determines whether or not the group becomes the primary group for the object, or is given private authority to the object. If the user who owns the object changes to a different user group, the original group profile still retains authority to any objects created.

Even if the Owner field in a user profile is *GRPPRF, the user must still have sufficient storage to hold a new object while it is being created. After it is created, ownership is transferred to the group profile. The MAXSTG parameter in the user profile determines how much auxiliary storage a user is allowed.

Evaluate the objects a user might create, such as query programs, when choosing between group and individual user ownership:

- If the user moves to a different department and a different user group, should the user still own the objects?
- Is it important to know who creates objects? The object authority displays show the object owner, not the user who created the object.

Note: The Display Object Description display shows the object creator.

If the audit journal function is active, a Create Object (CO) entry is written to the QAUDJRN audit journal at the time an object is created. This entry identifies the creating user profile. The entry is written only if the QAUDLVL system value specifies *CREATE and the QAUDCTL system value includes *AUDLVL.

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

A profile must be a group profile (have a *gid*) to be assigned as the primary group for an object. The same profile cannot be the owner of the object and its primary group. When a user creates a new object, parameters in the user profile control whether the user's group is given authority to the object and the type of authority given. The Group Authority Type (GRPAUTTYP) parameter in a user profile can be used to make the user's group the primary group for the object.

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

Working with Primary Group Authority

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

- Change Object Primary Group (CHGOBJPGP)
- Work with Objects by Primary Group (WRKOBJPGP)
- Change Primary Group (CHGPGP)

When you change an object's primary group, you specify what authority the new primary group has. You can also revoke the old primary group's authority. If you do not revoke the old primary group's authority, it becomes a private authority. The new primary group cannot be the owner of the object. To change an object's primary group, you must have all of the following:

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

Using a Referenced Object

Both the Edit Object Authority display and the GRTOBJAUT command allow you to give authority to an object (or group of objects) based on the authority of a referenced object. This is a useful tool in some situations, but you should also evaluate the use of an authorization list to meet your requirements.

Application description worksheet:

This worksheet should be completed for each application on your system.

Table 92. Application description worksheet

Application description worksheet	
Prepared by:	Date:
Instructions • Prepare a separate worksheet for each application. • You do not need to enter the information on this worksheet into the system.	
Application name:	Abbreviation:
Brief description of the application:	
Primary menu name:	Library:
Initial program name:	Library:
List the libraries used by the application for both files and programs:	
Define the security objectives for the application, such as whether any information is confidential:	

Plan application installation:

To finish planning resource security, you need to prepare for your application installation.

The following topics will help you plan ownership and authority to your applications after you install them. The methods described here may not work for all applications. Consult your programmer or application provider for help with developing an effective installation plan.

If you plan to acquire an application from an application provider, use this information to plan the security activities you need to do before and after you load the application libraries. If you plan to install an application that programmers developed on your own system, use this information to plan the security activities necessary to move the application from test to production status. Work through the steps with one application. Then go back and prepare Application installation forms for any additional applications.

Make a copy of the following forms and fill them in as you work through this information:
- Application description form, you will need to complete one per application
- Library description form
- Authorization list form

After you finished these planning tasks, you are ready to set up your user security.

Plan authorization lists

You can group objects with similar security requirements by using an authorization list.

Conceptually, an authorization list contains a list of users and the authority that the users have to the objects that are secured by the list. Authorization lists provide an efficient way to manage the authority to similar objects on the system. However, in some cases, they make it difficult to keep track of authorities to objects. You can use the Print Private Authority (PRTPVTAUT) command to print information about authorization list authorities.

Authorization List Security

You can group objects with similar security requirements using an authorization list. An authorization list, conceptually, contains a list of users and the authority that the users have to the objects secured by the list. Each user can have a different authority to the set of objects the list secures. When you give a

user authority to the authorization list, the operating system actually grants a private authority for that user to the authorization list. You can also use an authorization list to define public authority for the objects on the list. If the public authority for an object is set to *AUTL, the object gets its public authority from its authorization list.

The authorization list object is used as a management tool by the system. It actually contains a list of all objects which are secured by the authorization list. This information is used to build displays for viewing or editing the authorization list objects.

You cannot use an authorization list to secure a user profile or another authorization list. Only one authorization list can be specified for an object. Only the owner of the object, a user with all object (*ALLOBJ) special authority, or a user with all (*ALL) authority to the object, can add or remove the authorization list for an object.

Objects in the system library (QSYS) can be secured with an authorization list. However, the name of the authorization list that secures an object is stored with the object. In some cases, when you install a new release of the operating system, all the objects in the QSYS library are replaced. The association between the objects and your authorization list would be lost.

Planning Authorization Lists

An authorization list has these advantages:

- Authorization lists simplify managing authorities.
- User authority is defined for the authorization list, not for the individual objects on the list. If a new object is secured by the authorization list, the users on the list gain authority to the object.
- One operation can be used to give a user authority to all the objects on the list.
- Authorization lists reduce the number of private authorities on the system. Each user has a private authority to one object, the authorization list. This gives the user authority to all the objects on the list. Reducing the number of private authorities in the system has the following advantages:
  - Reducing the size of user profiles
  - Improves the performance when saving the system (SAVSYS) or saving security data (SAVSECDTA)
- Authorization lists provide a good way to secure files. If you use private authorities, each user will have a private authority for each file member. If you use an authorization list, each user will have only one authority. Also, files that are open cannot have authority granted to the file or revoked from the file. If you secure the file with an authorization list, you can change the authorities, even when the file is open.
- Authorization lists provide a way to remember authorities when an object is saved. When an object is saved that is secured by an authorization list, the name of the authorization list is saved with the object. If the object is deleted and restored to the same system, it is automatically linked to the authorization list again. If the object is restored on a different system, the authorization list is not linked, unless ALWOBJDIF(*ALL) is specified on the restore command.

Advantages of Using an Authorization List

From a security management view, an authorization list is the preferred method to manage objects that have the same security requirements. Even when there are only a few objects that would be secured by the list, there is still an advantage to using an authorization list instead of using private authorities on the authorized to the objects. It is also easier to secure any new objects with the same authorities as the existing objects.

If you use authorization lists, then you should not have private authorities on the object. Two searches of the user's private authorities are required during the authority checking if the object has private authorities and the object is also secured by an authorization list. The first search is for the private

authorities on the object; the second search is for the private authorities on the authorization list. Two searches require use of system resources and might impact performance.

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. 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. Use 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)`

Use authorization lists

iSeries Navigator provides security features designed to assist you in developing a security plan and policy, and configure your system to meet your company's needs. One of the functions available is the use of authorization lists. Authorization lists have the following features:

- An authorization list groups objects with similar security requirements.
- An authorization list conceptually contains a list of users and groups, and the authority each has to the objects secured by the list.
- Each user and group can have a different authority to the set of objects the list secures.
- Authority can be given by way of the list, rather than to individual users and groups. Tasks that can be done using authorization lists include:
  - Create an authorization list
  - Change an authorization list
  - Add users and groups
  - Change user permissions
  - Display secured objects
  -

To use this function, perform these steps:

1. From iSeries Navigator, expand your server—>Security. You will see Authorization Lists and Policies.
2. Right-click Authorization Lists and select New Authorization List. The New Authorization List allows you to:
   - Use: Allows access to the object attributes and use of the object. The public may view, but not change the objects.
   - Change: Allows the contents of the object to be changed, with some exceptions.
   - All: Allows all operations on the object, except those that are limited to the owner. The user or group can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user or group can also change ownership of the object.
   - Exclude: All operations on the object are prohibited. No access or operations are allowed to the object for the users and groups having this permission type. Specifies the public is not allowed to use the object.

When working with authorization lists you will want to grant permissions for both objects and data. Object permissions you can choose include:

- Operational: Provides the permission to look at the description of an object and use the object as determined by the data permission that the user or group has to the object.
- Management: Provides the permission to specify the security for the object, move or rename the object, and add members to the database files.
- Existence: Provides the permission to control the object's existence and ownership. The user or group can delete the object, free storage of the object, perform save and restore operations for the object, and transfer ownership of the object. If a user or group has special save permission, the user or group does not need object existence permission.
- Alter (used only for database files and SQL packages): Provides the permission needed to alter the attributes of an object. If the user or group has this permission on a database file, the user or group can add and remove triggers, add and remove referential and unique constraints, and change the attributes of the database file. If the user or group has this permission on an SQL package, the user or group can change the attributes of the SQL package. This permission is currently used only for database files and SQL packages.
- Reference (used only for database files and SQL packages): Provides the permission needed to reference an object from another object such that operations on that object may be restricted by the other object. If the user or group has this permission on a physical file, the user or group can add referential constraints in which the physical file is the parent. This permission is currently used only for database files. Data permissions you can choose are listed below.
- Read: Provides the permission needed to get and display the contents of the object, such as viewing records in a file.
- Add: Provides the permission to add entries to an object, such as adding messages to a message queue or adding records to a file.
- Update: Provides the permission to change the entries in an object, such as changing records in a file.
- Delete: Provides the permission to remove entries from an object, such as removing messages from a message queue or deleting records from a file.
- Execute: Provides the permission needed to run a program, service program or SQL package. The user can also locate an object in a library or directory.

For more information on each process as you are creating or editing your authorization lists, use the online help available in iSeries Navigator.

To simplify managing authorities, use an authorization list to group objects with the same requirements. You can then give the public, group profiles, and user profiles authority to the authorization list rather than to the individual objects on the list. The system treats every object that you secure by an authorization list the same, but you can give different users different authorities to the entire list.

An authorization list makes it easier to reestablish authorities when you restore objects. If you secure objects with an authorization list, the restore process automatically links the objects to the list. You can give a group or user the authority to manage an authorization list (*AUTLMGT). Authorization list management allows the user to add and remove other users from the list and to change the authorities for those users.

Recommendations:

- Use authorization lists for objects that require security protection and that have similar security requirements. Using authorization lists encourages you to think about categories of authority rather than individual authorities. Authorization lists also make it easier to restore objects and to audit the authorities on your system.
- Avoid complicated schemes that combine authorization lists, group authority, and individual authority. Choose the method that best suits the requirement, rather than using all of the methods at the same time.

Look at the group and individual authorities on your Library description forms. Decide if using authorization lists is appropriate. If so, prepare Authorization list forms and update your Library description forms with the authorization list information.

Related concepts

"Authorization lists" on page 9
Like a group profile, an authorization list allows you to group objects with similar security requirements and associate the group with a list of users and user authorities.

Authorization lists worksheet:

Use this worksheet to create an Authorization list for an application on your system.

Table 93. Authorization list worksheet

Authorization list worksheet					
Prepared by:			Date:		
Instructions • Prepare one worksheet for each authorization list. • Use the worksheet to list the objects that the list and the groups and individuals who have access to the list secure.					
Authorization list name:					
Description:					
List the objects which the list secures					
Object name	Object type	Object library	Object name	Object type	Object library
List groups and users who have access to the list					
Group or user	Type of access allowed	List management?	Group or user	Type of access allowed	List management?

Plan database file security

This topic discusses the steps necessary to create a security plan for your database files.

Structured Query Language (SQL) uses cross-reference files to keep track of database files and their relationships. These files are collectively referred to as the SQL catalog. Public authority to the SQL

catalog is *READ. This means that any user who has access to the SQL interface can display the names and text descriptions for all files on your system. The SQL catalog does not affect the normal authority required to access the contents of database files.

Care should be taken when using a CL program that adopts authority to start SQL or Query Manager. Both of these query programs allow users to specify a file name. The user can, therefore, access any file that the adopted profile has authority to.

Planning File Security

The information contained in database files is usually the most important asset on your system. Resource security allows you to control who can view, change, and delete information in a file. If users require different authority to files depending on the situation, you can use adopted authority. For critical files on your system, keep a record of what users have authority to the file.

If you use group authority and authorization lists, you need to keep track of users who have authority through those methods, as well as users who are directly authorized. If you use adopted authority, you can list programs that adopt the authority of a particular user using the Display Program Adopt (DSPPGMADP) command.

You can also use the journaling function on the system to monitor activity against a critical file. Although the primary intent of a journal is to recover information, it can be used as a security tool. It contains a record of who has accessed a file and in what way. You can use the Display Journal (DSPJRN) command to view a sampling of journal entries periodically.

Securing Logical Files

Resource security on the system supports field-level security of a file. You can also use logical files to protect specific fields or records in a file. 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.

Plan integrated file system security

The integrated file system provides you with multiple ways to store and view information on the server.

The integrated file system is a part of the i5/OS operating system that supports stream input and output operations. It provides storage management methods that are similar to (and compatible with) personal computer operating systems and UNIX® operating systems. With the integrated file system, all objects on the system can be viewed from the perspective of a hierarchical directory structure. However, in most cases, users view objects in the way that is most common for a particular file system. For example, ⌂traditional⌂ objects are in the QSYS.LIB file system. Typically, users view these objects from the perspective of libraries. Users typically view objects in the QDLS file system from the perspective of documents within folders. The root (/), QOpenSys, and user-defined file systems present a structure of hierarchical (nested) directories. As a security administrator, you need to understand the following:

- Which file systems are used on your system
- The unique security characteristics of each file system

The following information provides some general considerations for the security of the integrated file system.

The integrated file system approach to security

The root file system acts as an umbrella (or a foundation) for all other server file systems. At a high level, it provides an integrated view of all of the objects on the system. Other file systems that can exist on servers provide varying approaches to object management and integration, depending on the underlying purpose of each file system. The QOPT (optical) file system, for example, allows applications and servers (including the iSeries Access for Windows file server) to access the CD-ROM drive on the server. Similarly, the QFileSvr.400 file system allows applications to access integrated file system data on remote servers.

The security approach for each file system depends on the data that the file system makes available. The QOPT file system, for example, does not provide object-level security because no technology exists to write authority information to a CD-ROM. For the QFileSvr.400 file system, access control occurs at the remote system (where the files are physically stored and managed). For file systems like QLANSrv, the Integrated xSeries® Server for iSeries provides access control. Despite the differing security models, many file systems support consistent management of access control through the integrated file system commands, such as Change Authority (CHGAUT) and Change Owner (CHGOWN).

Here are some tips related to the nooks and crannies of integrated file system security. The integrated file system is designed to follow POSIX standards as closely as possible. This leads to some interesting behavior where server authority and POSIX permissions are ⌂blended⌂:

1. Do not remove the private authority for a user to a directory owned by that user, even if that user is authorized through the public authority, a group, or authorization list. When working with libraries or folders in the standard server security model, removing the owner's private authority would reduce the amount of authority information stored for a user profile and would not affect other operations. But, because of the way the POSIX standard defines permission inheritance for directories, the owner of a newly-created directory will have the same object authorities to that directory as the owner of the parent has to the parent, even if the owner of the newly-created directory has other private authorities to the parent.

   That may be difficult to understand, so here is an example: USERA owns directory /DIRA, but USERA's private authorities have been removed. USERB has private authority to /DIRA. USERB creates directory /DIRA/DIRB. Because USERA has no object authorities to /DIRA, USERB will have no object authorities to /DIRA/DIRB. USERB will be unable to rename or delete /DIRA/DIRB without further action to change USERB's object authorities. This also comes into play when creating files with the open() API using the O_INHERITMODE flag. If USERB created a file /DIRA/FILEB, USERB would have no object authorities AND no data authorities to it. USERB could not write to the new file.
2. Adopted authority is not honored by most physical file systems. This includes the root (/), QOpenSys, QDLS, and user-defined file systems.
3. Any objects are owned by the user profile which created the objects, even if the OWNER field of the user profile is set to *GRPPRF.
4. Many file system operations require *RX data authority to every component of the path, including the root (/) directory. When experiencing authority problems, make sure to check the user's authorization to the root itself.
5. Displaying or retrieving the current working directory (DSPCURDIR, getcwd(), etc.) requires *RX data authority to every component in the path. However, changing the current working directory (CD, chdir(), etc.) only requires *X data authority to every component. Therefore, a user may change the current working directory to a certain path and then be unable to display that path.
6. The intent of the COPY command is to duplicate an object. The authority settings on the new file will be the same as the original except for the owner. The intent of the CPYTOSTMF command, however, is simply to duplicate data. The authority settings on the new file cannot be controlled by the user. The creator/owner will have *RWX data authority, but the group and public authorities will be *EXCLUDE. The user must use another means (CHGAUT, chmod(), etc.) to assign the desired authorities.

7. A user must be the owner or have *OBJMGT object authority to an object to retrieve authority information about the object. This pops up in some unexpected places, like COPY, which must retrieve the authority information on the source object to set the equivalent authorities on the target object.
8. When changing the owner or group of an object, the user must not only have appropriate authority to the object, but also must have *ADD data authority to the new owner/group user profile and *DELETE data authority to the old owner/group profile. These data authorities are not related to the file system data authorities. These data authorities can be displayed using the DSPOBJAUT command and changed using the EDTOBJAUT command. This also pops up unexpectedly on COPY when it tries to set the group ID for a new object.
9. The MOV command is prone to puzzling authority errors, especially when moving from one physical file system to another, or when performing data conversion. In these cases, the move actually becomes a copy-and-delete operation. Therefore, the MOV command can be affected by all of the same authority considerations as the COPY command (see 7 and 8 above) and the RMVLNK command, in addition to other specific MOV considerations.

When using the integrated file system APIs, you can restrict access to objects as you can when using data management interfaces. Be aware, however, that adopting authorities is not supported. An integrated file system API uses the authority of the user profile under which the job is running.

Each file system may have its own special authority requirements. NFS server jobs are the only exception to this rule. Network File System server requests run under the profile of the user whose user identification (UID) number was received by the NFS server at the time of the request. Authorities on your server are the equivalent of permissions on UNIX® systems. The types of permissions are read and write (for a file or a directory) and execute (for a file) or search (for a directory).

The permissions are indicated by a set of permission bits, which make up the ″mode of access″ of the file or directory. You can change the permission bits by using the ″change mode″ functions chmod() or fchmod(). You can also use the umask() function to control which file permission bits are set each time a job creates a file.

Considerations for integrated file systems security:

The “root” (/) file system acts as an umbrella or a foundation for all other file systems on the server. At a high level, it provides an integrated view of all of the objects on the system.

Other file systems that can exist on iSeries servers provide varying approaches to object management and integration, depending on the underlying purpose of each file system. The QOPT (optical) file system, for example, allows iSeries applications and servers (including the iSeries Access for Windows file server) to access the CD-ROM drive on the iSeries server. Similarly, the QFileSvr.400 file system allows applications to access integrated file system data on remote iSeries servers. The QLANSrv file server allows access to files stored on Integrated xSeries Server for iSeries or other connected servers in the network.

The security approach for each file system depends on the data that the file system makes available. The QOPT file system, for example, does not provide object-level security because no technology exists to write authority information to a CD-ROM. For the QFileSvr.400 file system, access control occurs at the remote system, where the files are physically stored and managed. For file systems like QLANSrv, the Integrated xSeries Server for iSeries provides access control. Despite the differing security models, many file systems support consistent management of access control through the integrated file system commands, such as Change Authority (CHGAUT) and Change Owner (CHGOWN).

Here are some tips related to the intricacies of integrated file system security. The integrated file system is designed to follow POSIX standards as closely as possible. This leads to some interesting behavior where iSeries server authority and POSIX permissions are used together:

1. Do not remove the private authority for a user to a directory owned by that user, even if that user is authorized through the public authority, a group, or authorization list. When working with libraries or folders in the standard iSeries server security model, removing the owner's private authority would reduce the amount of authority information stored for a user profile and would not affect other operations. But, because of the way the POSIX standard defines permission inheritance for directories, the owner of a newly-created directory will have the same object authorities to that directory as the owner of the parent has to the parent, even if the owner of the newly-created directory has other private authorities to the parent. For example:

   USERA owns directory /DIRA, but USERA's private authorities have been removed. USERB has private authority to /DIRA. USERB creates directory /DIRA/DIRB. Because USERA has no object authorities to /DIRA, USERB will have no object authorities to /DIRA/DIRB. USERB will be unable to rename or delete /DIRA/DIRB without further action to change USERB's object authorities. This also comes into play when creating files with the open() API using the O_INHERITMODE flag. If USERB created a file /DIRA/FILEB, USERB would have no object authorities AND no data authorities to it. USERB could not write to the new file.
2. Adopted authority is not honored by most physical file systems. This includes the "root" (/), QOpenSys, QDLS, and user-defined file systems.
3. Any objects are owned by the user profile which created the objects, even if the OWNER field of the user profile is set to *GRPPRF.
4. Many file system operations require *RX data authority to every component of the path, including the "root" (/) directory. When experiencing authority problems, make sure to check the user's authorization to the "root" (/) directory.
5. Displaying or retrieving the current working directory (DSPCURDIR, getcwd(), etc.) requires *RX data authority to every component in the path. However, changing the current working directory (CD, chdir(), etc.) only requires *X data authority to every component. Therefore, a user may change the current working directory to a certain path and then be unable to display that path.
6. The intent of the COPY command is to duplicate an object. The authority settings on the new file will be the same as the original except for the owner. The intent of the CPYTOSTMF command, however, is simply to duplicate data. The authority settings on the new file cannot be controlled by the user. The creator/owner will have *RWX data authority, but the group and public authorities will be *EXCLUDE. The user must use another means (CHGAUT, chmod(), etc.) to assign the desired authorities.
7. A user must be the owner or have *OBJMGT object authority to an object to retrieve authority information about the object. This pops up in some unexpected places, like COPY, which must retrieve the authority information on the source object to set the equivalent authorities on the target object.
8. When changing the owner or group of an object, the user must not only have appropriate authority to the object, but also must have *ADD data authority to the new owner/group user profile and *DELETE data authority to the old owner/group profile. These data authorities are not related to the file system data authorities. These data authorities can be displayed using the DSPOBJAUT command and changed using the EDTOBJAUT command. This also pops up unexpectedly on COPY when it tries to set the group ID for a new object.
9. The MOV command is prone to puzzling authority errors, especially when moving from one physical file system to another, or when performing data conversion. In these cases, the move actually becomes a copy-and-delete operation. Therefore, the MOV command can be affected by all of the same authority considerations as the COPY command (see 7 and 8 above) and the RMVLNK command, in addition to other specific MOV considerations.

For more information about a specific file system on your iSeries server, you will need to consult the documentation for the licensed program that uses the file system.

Root, QOpenSys, and user-defined file systems:

These are security considerations for the root, QOpenSys, and user-defined file systems.

How authority works

The root, QOpenSys, and user-defined file systems provide a blending of iSeries server, PC, and UNIX** capabilities both for object management and for security. When you use the integrated file system commands from an iSeries server session (WRKAUT and CHGAUT), you can set all the normal iSeries server object authorities. This includes the *R, *W, and *X authorities that are compatible with Spec 1170 (UNIX-type operating systems).

Note: The root, QOpenSys, and user-defined file systems are functionally equivalent. The QOpenSys file system is case-sensitive. The root file system is not. User-defined file systems can be defined as case-sensitive. Because these file systems have the same security characteristics, you can assume in the topics that follow that their names are used interchangeably.

When you access the root file system as an administrator from a PC session, you can set object attributes that the PC uses to restrict certain types of access:

- System
- Hidden
- Archive
- Read-only

These PC attributes are in addition to, not replacements for, iSeries server object authority values.

When a user attempts to access an object in the root file system, OS/400 enforces all of the object authority values and attributes for the object, whether or not those authorities are ⌂visible⌂ from the user's interface. For example, assume that the read-only attribute for an object is set on. A PC user cannot delete the object through a iSeries Access interface. An iSeries server user with a fixed function workstation cannot delete the object either, even if the iSeries server user has *ALLOBJ special authority. Before the object can be deleted, an authorized user must use a PC function to reset the read-only value to off. Similarly, a PC user might not have sufficient OS/400 authority to change the PC-relevant security attributes of an object.

UNIX-type applications that run on iSeries servers use UNIX-like application programming interfaces (APIs) to access data in the root file system. With UNIX-like APIs, applications can recognize and maintain the following security information:

- Object owner
- Group owner (iSeries server primary group authority)
- Read (files)
- Write (change contents)
- Execute (run programs or search directories)
- S_ISVTX mode bit (restricted rename and unlink attribute)

The system maps these data authorities to existing iSeries server object and data authorities:

- Read (*R) = *OBJOPR and *READ
- Write (*W) = *OBJOPR, *ADD, *UPD, *DLT
- Execute (*X) = *OBJOPR and *EXECUTE

The concepts for other object authorities (*OBJMGT, *OBJEXIST, *OBJALTER, and *OBJREF) do not exist in a UNIX-type environment.

However, these object authorities do exist for all of the objects in the root file system. When you create an object using a UNIX-like API, that object inherits these authorities from the parent directory, resulting in the following:

- The new object's owner has the same object authority as the parent directory's owner.
- The new object's primary group has the same object authority as the parent directory's primary group.
- The new object's public has the same object authority as the parent directory's public.

The new object's data authority for owner, primary group, and public are specified on the API with the mode parameter. When all of the object authorities are set 'on', you get the authority behavior that you would expect in a UNIX-type environment. It is best to leave them set 'on', unless you do not want the POSIX-like behavior.

When you run applications that use UNIX-like APIs, the system enforces all object authorities, whether or not they are ⌂visible⌂ to UNIX-type applications. For example, the system will enforce the authority of authorization lists even though the concept of authorization lists does not exist in UNIX-type operating systems.

When you have a mixed-application environment, you need to ensure that you do not make authority changes in one environment that will break your applications in another environment.

Security commands for the root, QOpenSys, and user-defined file systems:

IBM provides a set of commands for working with objects in multiple file systems.

Commands

The following commands are related to system security:
- Change Auditing (CHGAUD)
- Change Authority (CHGAUT)
- Change Owner (CHGOWN)
- Change Primary Group (CHGPGP)
- Display Authority (DSPAUT)
- Work with Authority (WRKAUT)

In addition, UNIX-like APIs are available to work with security.

Authorities

***RW**
Read/write

***R** Read

***WX**
Write/execute

***W**
Write

***X** Execute

Public authority to the "root" directory:

When your system ships, the public authority to the "root" directory is *ALL (all object authorities and all data authorities).

This setting provides flexibility and compatibility with both what UNIX-like applications expect and what typical iSeries server users expect. An iSeries server user with command-line capability can create a new library in the QSYS.LIB file system simply by using the CRTLIB command. Normally, authority on a

typical iSeries server allows this. Similarly, with the shipped setting for the root file system, a typical user can create a new directory in the root file system (just like you can create a new directory on your PC).

As a security administrator, you must educate your users about adequately protecting the objects that they create. When a user creates a library, probably the public authority to the library should not be *CHANGE, the default value. The user should set public authority either to *USE or to *EXCLUDE, depending on the contents of the library.

If your users need to create new directories in the "root" (/), QOpenSys, or user-defined file systems, you have several security options:

- You can educate your users to override the default authority when they create new directories. The default is to inherit authority from the immediate parent directory. In the case of a newly created directory in the root directory, by default the public authority will be *ALL.
- You can create a master subdirectory under the "root" directory. Set the public authority on that master directory to an appropriate setting for your organization. Then instruct users to create any new personal directories in this master subdirectory. Their new directories will inherit its authority.
- You can consider changing the public authority for the "root" directory to prevent users from creating objects in that directory. You would do prevent users creating objects by removing *W, *OBJEXIST, *OBJALTER, *OBJREF, and *OBJMGT authorities. However, you need to evaluate whether this change will cause problems for any of your applications. You might, for example, have UNIX-like applications that expect to be able to delete objects from the "root" directory.

Restrict access to the QSYS.LIB file system:

You can use this information to restrict access to the QSYS.LIB file system.

Because the root file system is the umbrella file system, the QSYS.LIB file system appears as a subdirectory within the root directory. Therefore, any PC user with access to your server can manipulate objects stored in server libraries (the QSYS.LIB file system) with normal PC commands and actions. A PC user could, for example, drag a QSYS.LIB object (such as the library with your critical data files) to the shredder.

The system enforces all object authority whether or not it is visible to the interface. Therefore, a user cannot shred (delete) an object unless the user has *OBJEXIST authority to the object. However, if your system depends on menu access security rather than object security, the PC user might very well discover objects in the QSYS.LIB file system that are available for shredding.

As you expand the uses of your system and the different methods of access that you provide, you will soon discover that menu access security is not sufficient. However, servers also provide a simple way for you to prevent access to the QSYS.LIB file system through the root file system directory structure. You can use the QPWFSERVER authorization list to control which users can access the QSYS.LIB file system through the root directory.

When a user's authority to the QPWFSERVER authorization list is *EXCLUDE, the user cannot enter the QSYS.LIB directory from the root directory structure. When a user's authority is *USE, the user can enter the directory. Once the user has authority to enter the directory, normal object authority applies for any action the user attempts to perform on an object within the QSYS.LIB file system. In other words, the authority to the QPWFSERVER authorization list acts like a door to the entire QSYS.LIB file system. For the user with *EXCLUDE authority, the door is locked. For the user with *USE authority (or any greater authority), the door is open.

For most situations, users do not need to use a directory interface to access objects in the QSYS.LIB file system. Probably, you will want to set the public authority to the QPWFSERVER authorization list to *EXCLUDE. Keep in mind, that authority to the authorization list opens or closes the door to all libraries within the QSYS.LIB file system, including user libraries. If you encounter users who object to this

exclusion, you can evaluate their requirements on an individual basis. If appropriate, you can explicitly authorize an individual user to the authorization list. However, you need to ensure that the user has appropriate authority to objects within the QSYS.LIB file system. Otherwise, the user might unintentionally delete objects or entire libraries.

Note:

1. When your system ships, the public authority to the QPWFSERVER authorization list is *USE.
2. If you explicitly authorize an individual user, the authorization list controls access only with iSeries Access file serving, NetServer file serving and file serving between servers. This does not prevent access to the same directories via FTP, ODBC, and other networks.

Secure directories:

To access an object within the root file system, you read through the entire path to that object.

To search a directory, you must have *X (*OBJOPR and *EXECUTE) authority to that directory. Assume, for example, that you want to access the following object: /companya/customers/custfile.dat

You must have *X authority to the companya directory and to the customers directory.

With the root file system, you can create a symbolic link to an object. Conceptually, a symbolic link is an alias for a path name. Usually, it is shorter and easier to remember than the full path name. A symbolic link does not, however, create a different physical path to the object. The user still needs *X authority to every directory and subdirectory in the physical path to the object.

For objects in the root file system, you can use directory security just as you might use library security in the QSYS.LIB file system. You can, for example, set the public authority of a directory to *EXCLUDE to prevent public users from accessing any objects within that tree.

Security for new objects:

When you create a new object in the “root” (/) file system, the interface that you use to create it determines its authorities.

For example, if you use the CRTDIR command and its defaults, the new directory inherits all of the authority characteristics of its parent directory, including private authorities, primary group authority, and authorization list association. The following sections describe how authorities are determined for each type of interface.

Authority comes from the immediate parent directory, not from directories higher up in the tree. Therefore, as a security administrator, you need to view the authority that you assign to directories in a hierarchy from two perspectives:

- How the authority affects access to objects in the tree, like library authority.
- How the authority affects newly created objects, like the CRTAUT value for libraries.

Recommendation: You may want to give users who work in the integrated file system a home directory (for example, /home/usrxxx), then set the security appropriately, such as PUBLIC *EXCLUDE. Any directories the user creates under their home directory will then inherit the authorities.

Use the Create Directory command:

When you create a new subdirectory by using the CRTDIR command, you have two options for specifying authority.

These are the two options you have for specifying authority:

- You can specify the public authority. Public authority can be granted for data authority, object authority, or both.
- You can specify *INDIR for the data authority, object authority, or both. When you specify *INDIR for both data authority and object authority, the system makes an exact copy of all the authority information from the parent directory to the new object, including authorization list, primary group, public authority, and private authorities. The system does not copy private authority that the QSYS profile or the QSECOFR profile has to the object.

Create a directory with an API:

When you create a directory by using the mkdir() API, you specify the data authorities for the owner, the primary group, and public (using the authority map of *R, *W, and *X).

The system uses the information in the parent directory to set the object authorities for the owner, primary group, and public. Because UNIX-type operating systems do not have the concept of object authorities, the mkdir() API does not support specifying object authorities. If you want different object authorities, you can use the iSeries server command, CHGAUT. However, when you remove some object authorities, the UNIX-like application might not work as you expect it to work.

Create a stream file with the open() or creat() API:

When you use the creat() API to create a stream file, you can specify the data authorities for the owner, the primary group, and public (using the UNIX-like authorities of *R, *W, and *X).

The system uses the information in the parent directory to set the object authorities for the owner, primary group, and public. You can also specify these authorities when you use the open() API to create a stream file. Alternatively, when you use the open() API you can specify that the object should inherit all authorities from the parent directory. This is called inherit mode. When you specify inherit mode, the system then creates a complete match for the parent authorities, including authorization list, primary group, public authority, and private authorities. This option works like specifying *INDIR on the CRTDIR command.

Create an object by using a PC interface:

You can use the creat() API to create a stream file.

When you use the creat() API to create a stream file, you can specify the data authorities for the owner, the primary group, and public (using the UNIX-like authorities of *R, *W, and *X).

QFileSvr.400 file system:

With the QFileSvr.400 file system, a user (USERX) on one iSeries system (SYSTEMA) can access data on another connected iSeries system (SYSTEMB).

The USERX has an interface that is just like the Client Access interface. The remote iSeries server (SYSTEMB) appears as a directory with all its file systems as subdirectories. When USERX attempts to access SYSTEMB with this interface, SYSTEMA sends USERX's user profile name and encrypted password to SYSTEMB. The same user profile and password must exist on SYSTEMB or SYSTEMB rejects the request. If SYSTEMB accepts the request, USERX appears to SYSTEMB just like any Client Access user. The same authority-checking rules apply to any actions that USERX attempts.

As a security administrator, you need to be aware that the QFileSvr.400 file system represents another possible door to your system. You cannot assume that you are limiting your remote users to an interactive sign on with display station passthrough. If you have the QSERVER subsystem running and your system is connected to another iSeries system, remote users can access your system as if they are on

a local PC running Client Access. More than likely, your system will have a connection that needs to have the QSERVER subsystem running. This is yet another reason why a good object authority scheme is essential.

Network file system:

The Network File System (NFS) provides access to and from systems that have NFS implementations.

NFS is an industry-standard method for sharing information among users on networked systems. Most major operating systems (including PC operating systems) provide NFS. For UNIX systems, NFS is the primary method for accessing data. iSeries servers can act as both an NFS client and an NFS server.

When you are the security administrator of an iSeries system that acts as an NFS server, you need to understand and manage the security aspects of NFS. Suggestions and considerations:

- You must explicitly start the NFS server function by using the STRNFSSVR command. Control who has authority to use this command.
- You make a directory or an object available to NFS clients by exporting it. Therefore, you have very specific control over which parts of your system you will make available to NFS clients in your network.
- When you export, you can specify which clients have access to the objects. You identify a client by system name or IP address. A client can be an individual PC or an entire iSeries server or UNIX system. In NFS terminology, the client (IP address) is called a machine.
- When you export, you can specify read-only access or read/write access for each machine that has access to an exported directory or object. In most cases, you will probably want to provide read-only access.
- The NFS does not provide password protection. It is designed and intended for data sharing within a trusted community of systems. When a user requests access, the server receives the user's uid. Some uid considerations are:
  - The iSeries server attempts to locate a user profile with the same uid. If it finds a matching uid, it uses the credentials of the user profile. Credentials is an NFS term to describe using the authority of a user. This is similar to profile swapping in other iSeries server applications.
  - When you export a directory or object, you can specify whether you will allow access by a profile with root authority. The NFS server on iSeries servers equates root authority to *ALLOBJ special authority. If you specify that you will not allow root authority, an NFS user with a uid that maps to a user profile with *ALLOBJ special authority will not be able to access the object under that profile. Instead, if anonymous access is allowed, the requester will be mapped to the anonymous profile.
  - When you export a directory or object, you can specify whether you will allow anonymous requests. An anonymous request is a request with a uid that does not match any uid on your system. If you choose to allow anonymous requests, the system maps the anonymous user to the IBM-supplied QNFSANON user profile. This user profile does not have any special authorities or explicit authority. On the export, you can specify a different user profile for anonymous requests if you want.
- When your system participates in an NFS network, or any network with UNIX systems that depend on uids, you probably need to manage your own uids instead of letting the system assign them automatically. You will need to coordinate uids with other systems in your network.

  You might discover that you need to change uids, even for IBM-supplied user profiles, to have compatibility with other systems in your network. A program is available to make it simpler to change the uid for a user profile. When you change the uid for a user profile, you also need to change the uid for all the objects that the profile owns in either the root directory or the QOpenSrv directory. The QSYCHGID program automatically changes the uid in both the user profile and all the owned objects.

Plan printer and printer output queue security

This topic describes the key points in planning security for the printer and printer output queue, the importance of the planning tasks, and recommendations for completing the tasks.

Review the printer portion of your Physical Security Plan. Fill in the output queue section of the Printer Output and Workstation Security form as you work through this topic. You also need a plan to protect confidential information while it is printing or waiting to print. Check your Physical Security Plan for printers that your company uses for confidential output. After you plan printer output queue security, you can plan security for workstations.

The basic printing process involves the following key points:

- A copy of the report to be printed is held in a spooled file or printer output.
- The spooled file is stored in an object called an output queue until a printer is available.
- Spooling makes it easier to schedule printer jobs and to share printers.
- Spooling helps you protect confidential output.

You can create one or more special output queues to hold confidential output and restrict who can view and manage those output queues.

- To secure the special output queue, you can use these commands:
  - Work with Output Queue Description (WRKOUTQD)
  - Create Output Queue (CRTOUTQ)
  - Change Output Queue (CHGOUTQ)
- On these commands, you can specify values for these key parameters:
  - DSPDTA
  - AUTCHK
  - OPRCTL

When you run a program that prints a report, the report usually does not go directly to a printer. The program creates a copy of the report, called a spooled file or printer output. The system stores the spooled file in an object called an output queue until a printer is available. When the output queue contains printer output, you can view the report at your workstation. You can also hold it or direct it to a specific printer.

Spooling makes it easier to schedule printing jobs and to share printers. Spooling also helps you protect confidential output. You can create one or more special output queues to hold confidential output and restrict who can view and manage those output queues. You can also control when confidential output is sent from the queue to a printer. Complete the Printer Output and Workstation Security form as you work through this topic.

When you create a special output queue, you can specify several parameters that relate to security:

- **Display Data (DSPDTA) Parameter:** The DSPDTA parameter of an output queue determines whether a user can view, send, or copy a spooled file that another user owns.
- **Authority to Check (AUTCHK) Parameter:** The AUTCHK parameter specifies what type of authorities to the output queue allow the user to control all the files on the queue. Users with some special authority may also be able to control the files:
  - ***OWNER:** The requester must have ownership authority to the output queue in order to pass the output queue authorization test. The requester can have ownership authority by being the owner of the output queue, or sharing a group profile with the queue owner, or running a program that adopts the owner's authority.
  - ***DTAAUT:** Any user with add, read, and delete authority to the output queue can control all spooled files on the queue.
- **Operator Control (OPRCTL) Parameter:** The OPRCTL parameter of an output queue determines whether users with *JOBCTL special authority or *SYSOPR user class are allowed to control the output queue, provided that the profile was created with *SYSOPR user class, and that the special authorities parameter was set to *USRCLS and has not been changed.

The output queue parameters, the user's authority to the output queue, and the user's special authority work together to determine the functions a user can perform on spooled files in an output queue. You can perform the following printing functions with spooled files:

- Add spooled files to the queue.
- View a list of spooled files (WRKOUTQ command).
- Display, copy, or send spooled files (DSPSPLF, CPYSPLF, SNDNETSPLF, and SNDTCPSPLF commands).
- Change, delete, hold, or release spooled files (CHGSPLFA, DLTSPLF, HLDSPLF, and RLSSPLF commands).
- Change, clear, hold, and release output queue (CHGOUTQ, CLROUTO, HLDOUTQ, and RLSOUTQ commands).

For more information on the printing commands, see the following tables in "Appendix D" of *iSeries Security Reference*:

"Output Queue Commands"

"Spooled File Commands"

"Writer Commands"

Securing spooled files

A spooled file is a special type of object on the system. You cannot directly grant and revoke authority to view and manipulate a spooled file. The authority to a spooled file is controlled by several parameters on the output queue that holds the spooled file.

When you create a spooled file, you are the owner of that file. You can always view and manipulate any spooled files you own, regardless of how the authority for the output queue is defined. You must have *READ authority to add new entries to an output queue. If your authority to an output queue is removed, you can still access any entries you own on that queue using the Work with Spooled Files (WRKSPLF) command.

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. See Controlling printing to output queue or printer for more information.

If defaults are used, the default output queue of the printer device specified in the system value QPRTDEV printer is used.

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. A user with *SPLCTL cannot manipulate, display, or use spooled files on an iASP unless the user has authority to the iASP group. A user needs *EXECUTE authority to the primary iASP device description.

For more information on the following subjects, see "Printing" in Chapter 6 of the *iSeries Security Reference*:

"Display Data (DSPDTA) parameter of output queue"

"Authority to Check (AUTCHK) parameter of output queue"

"Operator Control (OPRCTL) parameter of output queue"

"Output queue and parameter authorities required for printing"

Examples: output queue

Following are several examples of setting security parameters for output queues to meet different requirements:

- Create a general purpose output queue. All users are allowed to display all spooled files. The system operators are allowed to manage the queue and change spooled files: `CRTOUTQ OUTQ(QGPL/GPOUTQ) DSPDTA(*YES) OPRCTL(*YES) AUTCHK(*OWNER) AUT(*USE)`
- Create an output queue for an application. Only members of the group profile GRPA are allowed to use the output queue. All authorized users of the output queue are allowed to display all spooled files. System operators are not allowed to work with the output queue: `CRTOUTQ OUTQ(ARLIB/AROUTQ) DSPDTA(*NO) OPRCTL(*NO) AUTCHK(*OWNER) AUT(*EXCLUDE)CHGOBJOWN 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 display, copy, send, or move other user's files 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, copy, send, or move other user's spooled files. `CRTOUTQ OUTQ(QGPL/CFOUTQ) DSPDTA(*OWNER) AUTCHK(*OWNER) OPRCTL(*YES) AUT(*USE)`

For more information, see Secure your printer output queue.

Worksheet needed: Printer output queue security worksheet

Printer output queue security worksheet:

Complete this worksheet as part of your printer output queue security.

Table 94. Printer output queue and workstation security worksheet

Printer output queue and workstation security worksheet				
Prepared by:		Date:		
Instructions • Make an entry on this worksheet for any workstation or output queue that requires special protection.				
List the parameters for restricted output queues				
Output queue name	Output queue library	Display any file (DSPDTA)	Authority to check (AUTCHK)	Operator control (OPRCTL)
Security officer workstations: If you limit the security officer to specific workstations, by setting the system value QLMTSECOFR is yes, list below the workstations authorized for the security officer and anyone with *ALLOBJ authority				

Table 94. Printer output queue and workstation security worksheet (continued)

Printer output queue and workstation security worksheet	
List below the authorities for restricted workstations	
Workstation name	Groups or users who are authorized (*CHANGE authority)
Note: Restricted workstations should have public authority set to *EXCLUDE.	

Plan workstation resource security

After planning resource security for printers and printer output, use this topic to begin planning workstation security.

On your Physical Security Plan, you listed workstations that represent a security risk because of their location. Use this information to determine which workstations you need to restrict.

You can encourage the people who use these workstations to be particularly aware of security. They should sign off whenever they leave their workstations. You may want to record your decision about sign off procedures for vulnerable workstations in your security policy. You can also limit which functions can be performed at those workstations to minimize the risks.

The easiest method for limiting function at a workstation is to restrict it to user profiles with limited function. You may choose to prevent people with security officer or service authority from signing on at every workstation. If you use the QLMTSECOFR system value to do this, people with security officer authority can sign on only at specifically authorized workstations.

Prepare the workstation portion of the Output queue and workstation security form. You should also review a list of resource security recommendations to ensure that your resource security plan is simple and complete. After you have reviewed the example and the recommendations you can begin planning your application installation.

Workstation security worksheet:

Complete this worksheet as you develop your workstation security plan.

Table 95. Workstation Security worksheet

Workstation Security worksheet	
Prepared by:	Date:
Instructions • Make an entry on this worksheet for any workstation that requires special protection.	
Security officer workstations: If you limit the security officer to specific workstations (system value QLMTSECOFR is yes), list below the workstations authorized for the security officer and anyone with *ALLOBJ authority:	
List below the authorities for restricted workstations:	
Workstation name	Groups or users who are authorized (*CHANGE authority)
Note: Restricted workstations should have public authority set to *EXCLUDE.	

Plan 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.

Guidelines for 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 required to perform the jobs or tasks assigned to the programmer.
- Do not use the QPGMR user profile as a group profile for programmers. Use test libraries and prevent access to production libraries.
- Create programmer libraries and use a program that adopts authority to copy selected production data to programmer libraries for testing.
- If interactive performance is an issue, consider changing the commands for creating programs to run only in batch: CHGCMD CMD(CRTxxxPGM) ALLOW(*BATCH *BPGM)
- Perform security auditing of application function before moving applications or program changes from test to production libraries.
- Use the group profile technique when an application is being developed. Have all application programs owned by a group profile. Make programmers who work on the application members of the group and define the programmer user profiles to have the group own any new objects created (OWNER(*GRPPRF)). When a programmer moves from one project to another, you can change the group information in the programmer's profile. See Group Ownership of Objects for more information.
- Develop a plan for assigning ownership of applications when they are moved into production. To control changes to a production application, all application objects, including programs, should be owned by the user profile designated for the application.

  Application objects should not be owned by a programmer because the programmer will have uncontrolled access to them in a production environment. The profile that owns the application may be the profile of the individual responsible for the application, or it may be a profile specifically created as the application owner.

Managing Source Files

Source files are important to the integrity of your system. They may also be a valuable company asset if you have developed or acquired custom applications. Source files should be protected like any other important file on the system. Place source files in separate libraries and controlling who can update them and move them to production.

When a source file is created on the system, the default public authority is *CHANGE, which allows any user to update any source member. By default, only the owner of the source file or a user with *ALLOBJ special authority can add or remove members. In most cases, this default authority for source physical files should be changed. Programmers working on an application need *OBJMGT authority to the source files to add new members. The public authority should probably be reduced to *USE or *EXCLUDE, unless the source files are in a controlled library.

Planning Security for System Programmers or Managers

Most systems have someone responsible for housekeeping functions. This person monitors the use of system resources, particularly disk storage, to make sure that users regularly remove unused objects to

free space. System programmers need broad authority to observe all the objects on the system. However, they do not need to view the contents of those objects.

You can use adopted authority to provide a set of display commands for system programmers, rather than giving special authorities in their user profiles.

Plan network security

When connecting to an untrusted network, your security policy must describe a comprehensive security scheme, including the security measures that you will implement at the network level.

Installing a firewall is one of the best means of deploying a comprehensive set of network security measures. Also, your Internet Service Provider (ISP) can and should provide an important element in your network security plan. Your network security scheme should outline what security measures your Internet Service Provider (ISP) will provide, such as filtering rules for the ISP router connection and public Domain Name Service (DNS) precautions. Continue to check with your ISP periodically to ensure they are continually upgrading their security measures, this will also help you keep your security plans current.

Although a firewall certainly represents one of your main lines of defense in your total security plan, it should not be your only line of defense. Because Internet security risks occur at a variety of levels, you need to set up security measures that provide multiple layers of defense against these risks.

While a firewall provides a tremendous amount of protection from certain kinds of attack, a firewall is only part of your total security solution. For instance, a firewall cannot necessarily protect data that you send over the Internet through applications such as SMTP mail, FTP, and TELNET. Unless you choose to encrypt this data, anyone on the Internet can access it as it travels to its destination.

Choosing network security options

Network security solutions that guard against unauthorized access generally rely on firewall technologies to provide the protection. To protect your system, you can choose to use a full-capability firewall product or you can choose to put into effect specific network security technologies as part of the i5/OS TCP/IP implementation. This implementation consists of the Packet rules feature, which includes IP filtering and NAT, and the HTTP for iSeries proxy server feature.

Choosing to use either the Packet rules feature or a firewall depends on your network environment, access requirements, and security needs. You should strongly consider using a firewall product as your main line of defense whenever you connect your system or your internal network to the Internet or other untrusted network.

A firewall is preferable in this case because a firewall typically is a dedicated hardware and software device with a limited number of interfaces for external access. When you use the i5/OS TCP/IP technologies for Internet access protection you are using a general purpose computing platform with a myriad number of interfaces and applications open to external access.

The difference is important for a number of reasons. For example, a dedicated firewall product does not provide any other functions or applications beyond those that comprise the firewall itself. Consequently, if an attacker successfully circumvents the firewall and gains access to the system, the attacker can not do much. Whereas, if an attacker circumvents the TCP/IP security functions on your system, the attacker potentially could have access to a variety of useful applications, services, and data. The attacker can then use these to wreck havoc on the system itself or to gain access to other systems in your internal network.

So, is it ever acceptable to use the TCP/IP security features? As with all the security choices that you make, you must base your decision on the cost versus benefit trade-offs that you are willing to make. You must analyze your business goals and decide what risks you are willing to accept versus the cost of how

you provide security to minimize these risks. The following table provides information about when it is appropriate to use TCP/IP security features versus a fully functional firewall device. You can use this table to determine whether you should use a firewall, TCP/IP security features, or a combination of both to provide your network and system protection.

Security technology	Best use of i5/OS TCP/IP technology	Best use of a fully functional firewall
IP packet filtering	• To provide additional protection for a single system, such as a public web server or an intranet system with sensitive data. • To protect a subnetwork of a corporate intranet when the system is acting as a gateway (casual router) to the rest of the network. • To control communication with a somewhat trusted partner over a private network or extranet where the system is acting as a gateway.	• To protect an entire corporate network from the Internet or other untrusted network to which your network is connected. • To protect a large subnetwork with heavy traffic from the remainder of a corporate network.
Network Address Translation (NAT)	• To enable the connection of two private networks with incompatible addressing structures. • To hide addresses in a subnetwork from a less trusted network.	• To hide addresses of clients accessing the Internet or other untrusted network. To use as an alternative to Proxy and SOCKS servers. • To make services of a system in a private network available to clients on the Internet.
Proxy server	To proxy at remote locations in a corporate network when a central firewall provides access to the Internet.	To proxy an entire corporate network when accessing the Internet.

Plan network attributes

If you have primarily NetWare servers on your network, you can simplify working with those servers by changing default values on a system-wide basis.

Many network server commands, such as DSPNWSUSR and WRKNWSSTS, allow you to specify *NWSA for a given parameter to indicate that the server should use information from the network server attributes.

For example, if you plan to enroll most of your users on the same set of NDS trees, you can simplify enrollment by first defining a default list of those trees. Then when you enroll users, you can refer to that list of default attributes by specifying *NWSA on the appropriate command parameters. Adding or removing network servers is also simpler because you change the default server list instead of manually changing all the profiles that refer to it.

When you are running TCP/IP, you must use the CHGNWSA command to add the TCP/IP names of the NetWare servers. Enhanced Integration for Novell NetWare uses this list of names to find the TCP/IP NetWare servers. This is the only place that the server uses the TCP/IP name of the NetWare server. After identifying NetWare servers from this list, NetWare Enhanced Integration knows the servers by their NetWare server names, not their TCP/IP names.

In addition, you can change the default value of the TCP/IP port from 20199 to some other value. If you change the default port value, you must load the NetWare Enhanced Integration NLM with the

parameter /tcp=*nnnn*, where *nnnn* is the new port value. If you decide to change this value after loading the NLM, you must unload and reload the NLM with the new value.

To set these attributes on an individual user profile basis instead, you can use the CHGNWSUSRA command. Network server attributes are saved by the Save System (SAVSYS) command. Network server attributes are restored to the system when the operating system is installed.

Network attributes describe the local system name, the default local location name, the default control point name, the local network identifier, and the network node type. If the machine is an end node, the attributes also contain the names of the network servers that are used by this system. Network attributes also determine whether the system uses HPR, or whether you want to use virtual controllers for APPN.

The Change Network Attributes (CHGNETA) command is used to set the attributes for the system within the network. The following attributes are defined for DISTRIB and these attributes apply to all connections in the network for this end node.

The network server user attributes preserve network information for a group or user profile. Many of the administrative commands use some of this information, such as the default server type, default context, and default NDS tree. The network server user attributes also contain a list of NDS trees, and associated user information that are used by the user enrollment support to enroll the user or group on NetWare. You can set defaults for this same information on a system-wide basis by using the Change Network Server Attribute (CHGNWSA) command. To specify these attributes on an individual or group profile basis and enroll server users to NetWare servers, you use the CHGNWSUSRA command. You use these attributes to specify the NDS trees on which you want to enroll users.

Network Print Server objects have attributes. The Network Print Server supports the following attributes. Refer to the data stream description for each object and action to determine the attributes that are supported for that combination.

Related information

Define network server attributes

Change network attributes

Change the network attributes (Distribution)

Network server user attributes

iSeries objects attributes

Plan APPC security

Use this information to understand how Advanced Program-to-Program Communication (APPC) works and how you can set up the appropriate security for APPC on your system.

APPC allows programs on i5/OS to communicate with programs on other systems having compatible communications support. Display station passthrough, distributed data management, personal computers, and iSeries Access for Windows can use APPC communications.

When your system participates in a network with other systems, a new set of doors and windows to your system becomes available. As security administrator, you should be aware of the options that you can use to control the entrances to your system in an APPC environment.

Tip: Many methods for connecting PCs to your system servers depend on communications, such as APPC or TCP/IP. Be sure to consider the security issues for connecting to both other systems and to PCs. When you plan your network protection, make sure that you do not adversely affect the PCs that are attached to your system.

These links provide additional information:

- APPC Programming

- APPC, APPN, and HPR

Example: A basic APPC session

In an APPC environment, when a user or application on one system requests access to another system, the two systems set up a session. To establish the session, the systems must link two matching APPC device descriptions.

The remote location name (RMTLOCNAME) parameter in the SYSTEMA device description must match the local location name (LCLLOCNAME) parameter in the SYSTEMB device description and vice versa. For the two systems to establish an APPC session, the location passwords in the APPC device descriptions on SYSTEMA and SYSTEMB must be identical. Both must specify *NONE, or both must specify the same value.

If the passwords are a value other than *NONE, they are stored and transmitted in encrypted format. If the passwords match, the systems establish a session. If the passwords do not match, the user's request is rejected.

Basic elements of APPC communications

APPC provides the ability for a user on one system to perform work on another system.

The system from which the request starts is called any of the following: **source system**, **local system**, or **client**.

The system that receives the request is called any of the following: **target system**, **remote system**, or **server**.

From the perspective of a security administrator, the following must happen before a user on one system (SYSTEMA) can perform meaningful work on another system (SYSTEMB):

- The source system (SYSTEMA) must provide a path to the target system (SYSTEMB). This path is called an **APPC session**.
- The target system must identify the user and associate the user with a user profile. The target system must support the encryption algorithm of the source system.
- The target system must start a job for the user with an appropriate environment (work management values).

The security administrator on the target system has primary responsibility for ensuring that APPC users do not violate security. However, when the security administrators on both systems work together, the job of managing APPC security is much easier.

APPC user access to the target system

The topics that follow describe the elements that determine how an APPC user gains entrance to the target system.

When the systems establish the APPC session, they create a path for the requesting user to get to the door of the target system. The server associates a user ID with a request for an APPC session. Several other elements determine what the user must do to gain entrance to the other system.

System methods for sending information about a user:

APPC architecture provides three methods for sending security information about a user from the source system to the target system.

These methods are referred to as the architected security values. The APPC Programming book provides more information about the architected security values.

The following table shows the security values in the APPC architecture:

Table 96. Security values in the APPC architecture

Architected security value	User ID set to target system	Password sent to target system
None	No	No
Same	Yes[1]	See note 2.
Program	Yes	Yes[3]

Note:

1. The source system sends the user ID if the target system specifies SECURELOC(*YES) or SECURELOC(*VFYENCPWD).
2. The user does not enter a password on the request because the password is already verified by the source system. For SECURELOC(*YES) and SECURELOC(*NO), the source system does not send the password. For SECURELOC(*VFYENCPWD), the source system retrieves the stored, encrypted password and sends it, in encrypted form.
3. The system sends the password in encrypted form if both the source and target systems support password encryption. Otherwise, the password is not encrypted.

The application that you request determines the architected security value. For example, SNADS always uses SECURITY(NONE). DDM uses SECURITY(SAME). With display station passthrough, you specify the security value by using parameters on the STRPASTHR command.

In all cases, the target system chooses whether to accept a request with the security value that is specified on the source system. In some situations, the target system might reject the request completely. In other situations, the target system might force a different security value. For example, when you specify both a user ID and a password on the STRPASTHR command, the request uses SECURITY(PGM). However, if the QRMTSIGN system value is *FRCSIGNON on the target system, you still see a Sign On display. With the *FRCSIGNON setting, the systems always use SECURITY(NONE), which is the equivalent of the user entering no user ID and password on the STRPASTHR command.

The source and target systems negotiate the security value before data is sent. In the situation where the target system specifies SECURELOC(*NO) and the request is SECURITY(SAME), for example, the target system tells the source system to use SECURITY(NONE). The source system does not send the user ID.

The target system rejects a session request when the user's password on the target system has expired. This applies only to connection requests that send a password, including the following:

- Session requests of type SECURITY(PROGRAM).
- Session requests of type SECURITY(SAME) when the SECURELOC value is *VFYENCPWD.

Options for dividing network security responsibility:

When your system participates in a network, you must decide whether to trust the other systems to validate the identity of a user who is trying to enter your system.

Will you trust SYSTEMA to ensure that USERA is really USERA (or QSECOFR is really QSECOFR)? Or will you require a user to provide a user ID and password again?

The secure location (SECURELOC) parameter on the APPC device description on the target system specifies whether the source system is a secure (trusted) location.

When both systems are running a release that supports *VFYENCPWD, SECURELOC(*VFYENCPWD) provides additional protection when applications use SECURITY(SAME). Although the requester does not enter a password on the request, the source system retrieves the user's password and sends it with the request. For the request to be successful, the user must have the same user ID and password on both systems.

When the target system specifies SECURELOC(*VFYENCPWD) and the source system does not support this value, the target system handles the request as SECURITY(NONE).

Table 97. How the APPC security value and the SECURELOC value work together

Source system	**Target system**	
Architected security value	**SECURELOC value**	**User profile for job**
None	Any	Default user[1]
Same	*NO	Default user[1]
	*YES	Same user profile name as requester from source system
	*VFYENCPWD	Same user profile name as requester from source system. The user must have the same password on both systems.
Program	Any	The user profiles that are specified on the request from the source system
Note: 1. The default user is determined by the communications entry in the subsystem description.		

Plan TCP/IP security

TCP/IP (Transmission Control Protocol/Internet Protocol) is a common way that computers of all types communicate with each other.

TCP/IP applications are well-known and widely used throughout the Internet. This topic provides tips for the following:

- Preventing TCP/IP applications from running on your system.
- Protecting system resources when you allow TCP/IP applications to run on your system.

SecureWay describes security considerations when you connect your iSeries server either to the Internet (a very large TCP/IP network) or to an intranet. Keep in mind that iSeries servers support many possible TCP/IP applications. When you decide to allow one TCP/IP application on your system, you may also be enabling other TCP/IP applications. As security administrator, you need to be aware of the range of TCP/IP applications and the security implications of these applications.

TCP/IP security components

You can take advantage of several TCP/IP security components that enhance your network security and add flexibility.

Though some of these technologies are also found in firewall products, these TCP/IP security components for i5/OS are not intended to be used as a firewall. However, you may be able to use some of these features, in some instances to eliminate the need for a separate firewall product. You also may be able to use these TCP/IP features to provide additional security in environments where you already use a firewall.

The following components can be utilized to enhance TCP/IP Security:

- Packet Rules
- HTTP Proxy Server
- VPN (virtual private networking)
- SSL (secure sockets layer)

Use packet rules to secure TCP/IP traffic:

Packet rules, which are the combination of IP filtering and network address translation (NAT), act like a firewall to protect your internal network from intruders.

IP filtering lets you control what IP traffic to allow into and out of your network. Basically, it protects your network by filtering packets according to rules that you define. NAT, on the other hand, allows you to hide your unregistered private IP addresses behind a set of registered IP addresses. This helps to protect your internal network from outside networks. NAT also helps to alleviate the IP address depletion problem, since many private addresses can be represented by a small set of registered addresses.

HTTP proxy server:

The HTTP proxy server comes with the IBM HTTP Server for iSeries server.

The HTTP Server is part of i5/OS. The proxy server receives HTTP requests from Web browsers and resends them to Web servers. Web servers that receive the requests are only aware of the IP address of the proxy server and cannot determine the names or addresses of the PCs that originated the requests. The proxy server can handle URL requests for HTTP, FTP, Gopher and WAIS.

The proxy server caches returned Web pages from requests made by all proxy server users. Consequently, when users request a page, the proxy server checks whether the page is in the cache. If it is, the proxy server returns the cached page. By using cached pages, the proxy server is able to server Web pages more quickly, which eliminates potentially time-consuming requests to the Web server. The proxy server can also log all URL requests for tracking purposes. You can then review the logs to monitor use and misuse of network resources.

You can use the HTTP proxy support in the IBM HTTP Server to consolidate Web access. Addresses of PC clients are hidden from the Web servers they access; only the IP address of the proxy server is known. Web page caching can also reduce communication bandwidth requirements and firewall workload.

Virtual private networking:

A virtual private network (VPN) allows your company to securely extend its private intranet over the existing framework of a public network, such as the Internet.

With VPN, your company can control network traffic while providing important security features such as authentication and data privacy. i5/OS VPN is an optionally-installable component of iSeries Navigator, the graphical user interface (GUI) for i5/OS. It allows you to create a secure end-to-end path between any combination of host and gateway. i5/OS VPN uses authentication methods, encryption algorithms, and other precautions to ensure that data sent between the two endpoints of its connection remains secure.

VPN runs on the network layer of the TCP/IP layered communications stack model. Specifically, VPN uses the IP Security Architecture (IPSec) open framework. IPSec provides base security functions for the Internet, as well as furnishes flexible building blocks from which you can create robust, secure virtual private networks. VPN also supports Layer 2 Tunnel Protocol (L2TP) VPN solutions. L2TP connections, which are also called virtual lines, provide cost-effective access for remote users by allowing a corporate network server to manage the IP addresses assigned to its remote users. Further, L2TP connections provide secure access to your system or network when you protect them with IPSec.

It is important that you understand the impact a VPN will have on your entire network. Proper planning and implementation are essential to your success. You should review the VPN topic in the iSeries Information Center to ensure that you know how VPNs work and how you might use them.

Secure sockets layer:

Secure Sockets Layer (SSL) has become an industry standard for enabling applications for secure communication sessions over an unprotected network, such as the Internet.

The SSL protocol establishes a secure connections between clients and server applications which provide authentication of one or both endpoints of the communication session. SSL also provides privacy and integrity of the data that client and server applications exchange. For more information, see Secure Sockets Layer.

Secure your TCP/IP environment

This topic provides general suggestions for steps that you can take to reduce the security exposures in the TCP/IP environment on your system.

These tips apply to your entire TCP/IP environment rather than to the specific applications that are discussed in the topics that follow:

- When you write an application for a TCP/IP port, make sure that the application is properly secure. You should assume that an outsider might try to access that application through that port. A knowledgeable outsider may attempt to TELNET to that application.
- Monitor the use of TCP/IP ports on your system. A user application that is associated with a TCP/IP port can provide "back-door" entry to your system without a user ID or a password. Someone with sufficient authority on your system can associate an application with a TCP or UDP port.
- As a security administrator, you should be aware of a technique called IP spoofing that is used by hackers. Every system in a TCP/IP network has an IP address. Someone who uses IP spoofing sets up a system (usually a PC) to pretend to be an existing IP address or a trusted IP address. Thus, the imposter can establish a connection with your system by pretending to be a system that you normally connect with.

  If you run TCP/IP on your system and your system participates in a network that is not physically protected, such as all nonswitched lines and predefined links, you are vulnerable to IP spoofing. To protect your system from damage by a "spoofer," start with the suggestions in this chapter, for example, sign-on protection and object security. You should also ensure that your system has reasonable auxiliary storage limits set. This prevents a spoofer from flooding your system with mail or spooled files to the point that your system becomes inoperable. In addition, you should regularly monitor TCP/IP activity on your system. If you detect IP spoofing, you can try to discover the weak points in your TCP/IP setup and to make adjustments.

  For your intranet, your company's private network of systems that do not need to connect directly to the outside, use IP addresses that are reusable. Reusable addresses are intended for use within a private network. The Internet backbone does not route packets that have a reusable IP address. Therefore, reusable addresses provide an added layer of protection inside your firewall. TCP/IP Setup provides more information about how IP addresses are assigned and about the ranges of IP addresses, as well as security information about TCP/IP.

Control which TCP/IP servers start automatically:

As security administrator, you need to control which TCP/IP applications start automatically when you start TCP/IP.

Commands for starting TCP/IP

Two commands are available for starting TCP/IP. For each command, the system uses a different method to determine which applications or servers to start.

STRTCP *Start TCP/IP*

The system starts every server that specifies AUTOSTART(*YES). Security recommendations:

- Assign *IOSYSCFG special authority carefully to control who can change the autostart settings.
- Carefully control who has authority to use the STRTCP command. The default public authority for the command is *EXCLUDE.

- Set up object auditing for the Change server-name Attributes commands (such as CHGTELNA) to monitor users who attempt to change the AUTOSTART value for a server.

STRTCPSVR *Start TCP/IP Server*

You use a parameter to specify which servers to start. The default when this command ships is to start all servers.

Security recommendations:

- Use the Change Command Default (CHGCMDDFT) command to set up the STRTCPSVR command to start only a specific server. This does not prevent users from starting other servers. However, by changing the command default, you make it less likely that users will start all servers by accident. For example, use the following command to set the default to start only the TELNET server:`CHGCMDDFT CMD(STRTCPSVR) NEWDFT('SERVER(*TELNET)')`

  Note: When you change the default value, you can specify only a single server. Choose either a server that you use regularly or a server that is least likely to cause security exposures (such as TFTP).
- Carefully control who has authority to use the STRTCPSVR command. The default public authority for the command is *EXCLUDE.

Table 98.

Server	Default value	Your value
Telnet	AUTOSTART(*YES)	
FTP (file transfer protocol)	AUTOSTART(*YES)	
BOOTP (bootstrap protocol)	AUTOSTART(*NO)	
TFTP (trivial file transfer protocol	AUTOSTART(*NO)	
REXEC (remote EXECution server)	AUTOSTART(*NO)	
RouteD (route daemon)	AUTOSTART(*NO)	
SMTP (simple mail transfer protocol)	AUTOSTART(*YES)	
POP (post office protocol)	AUTOSTART(*NO)	
HTTP (hypertext transfer protocol)[1]	AUTOSTART(*NO)	
ICS (Internet connection server)	AUTOSTART(*NO)	
LPD (line printer daemon)	AUTOSTART(*YES)	
SNMP (simple network management protocol)	AUTOSTART(*YES)	
DNS (domain name system)	AUTOSTART(*NO)	
DHCP (dynamic host configuration protocol)	AUTOSTART(*NO)	
NSMI	AUTOSTART(*NO)	
INETD	AUTOSTART(*NO)	
Note: 1. With the IBM HTTP Server, you use the CHGHTTPA command to set the AUTOSTART value.		

Prevent TCP/IP processing:

TCP/IP server jobs run in the QSYSWRK subsystem. You use the Start TCP/IP (STRTCP) command to start TCP/IP on your system.

If you do not want any TCP/IP processing or applications to run, do not use the STRTCP command. Your system ships with the public authority for the STRTCP command set to *EXCLUDE.

If you suspect that someone with access to the command is starting TCP/IP (during off-hours, for example), you can set up object auditing on the STRTCP command. The system will write an audit journal entry whenever a user runs the command.

Use the Secure Shell to secure your applications

You can set up the Secure Shell (SSH) to ensure the security of applications that run on the TCP/IP network.

TCP/IP connectivity applications, such as Telnet and FTP transmit data and passwords over the network in plain text. This means that the data and passwords can be intercepted and read by other users on the network.

The Secure Shell (SSH) protocol suite provides secure alternatives for Telnet and FTP. SSH verifies the authenticity of both the client and server. All of the data, including userids and passwords, is encrypted as it travels on the network.

For more information on using SSH, visit the following Web site: Portable Utilities for i5/OS.

Plan backup and recovery of security information

This information explains the necessity of planning the backup and recovery of your security information.

Saving your security information is just as important as saving your data. In some situations, you may need to recover user profiles, object authorities, and the data on your system. If you do not have your security information saved, you may need to manually rebuild user profiles and object authorities. This can be time-consuming and can lead to errors and security exposures. Planning adequate backup and recovery procedures for security information requires understanding how the information is stored, saved, and restored.

This table shows the commands used to save and restore security information. The sections that follow discuss saving and restoring security information in more detail.

Table 99. Commands for saving and restoring security information

	Save and Restore commands used				
Security information saved or restored	**SAVSECDTA SAVESYS**	**SAVCHGOBJ SAVOBJ SAVLIB SAVDLO SAVCFG**	**RSTUSRPRF**	**RSTOBJ RSTLIB RSTDLO RSTCFG**	**RSTAUT**
User profiles	X		X		
Object ownership[1]		X		X	
Primary group[1]		X		X	
Public authorities[1]		X		X	
Private authorities	X				X
Authorization lists	X		X		
Authority holders	X		X		
Link with the authorization list and authority holders		X		X	
Object auditing value		X		X	`
Function registration information[2]		X		X	

Table 99. Commands for saving and restoring security information (continued)

Security information saved or restored	Save and Restore commands used				
		SAVCHGOBJ			
		SAVOBJ		RSTOBJ	
		SAVLIB		RSTLIB	
	SAVSECDTA	SAVDLO		RSTDLO	
	SAVESYS	SAVCFG	RSTUSRPRF	RSTCFG	RSTAUT
Function usage information	X		X		X

[1] The SAVSECDTA, SAVSYS, and RSTUSRPRF commands save and restore ownership, primary group, primary group authority, and public authority for these object types: User profile (*USRPRF), Authorization list (*AUTL), and Authority holder (*AUTHLR).

[2] The object to save/restore is QUSEXRGOBJ, type *EXITRG in QUSRSYS library.

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.

Recovering your system often requires restoring data and associated security information. The usual sequence for recovery is:

1. Restore user profiles and authorization lists (RSTUSRPRF USRPRF(*ALL)).
2. Restore objects (RSTLIB, RSTOBJ, or RSTCFG).
3. Restore the private authorities to objects (RSTAUT).

Related information

Backup and Recovery PDF

Implement your security strategy

This topic describes the tasks for implementing your security strategy, explains why they are important, and provides links to the implementation topics.

This topic guides you through the tasks necessary to implement your security strategy. If you are setting up a new system, you should complete these steps in sequence. The system uses information from each step as you proceed to the next step. Setting up basic system security involves defining your user security, setting up system-level security, protecting your resources on the system, and setting up network security. The tables below highlight each of the steps you must configure to set up user and resource security.

Before you begin

If you are installing a new system, do these things before you start setting up security:

1. Make sure your system unit and your devices are installed and working properly. If you do not plan to use system naming for your devices, wait to attach your workstations and printers until after you change the system value that determines how devices are named (QDEVNAMING). Applying the new system values tells you when to attach the devices.
2. Load any licensed programs you plan to use.

Note: You *must* complete all the steps to set up user security first, before you begin setting up resource and network security.

Table 100. Steps in setting up system security

Step	What you do in this step	What worksheets you use
Set up your user environment.	Set up initial system values and network attributes.	System values selection
Set up system-level security	Set up additional system values.	 Security planner

Table 101. Steps in setting up resource security

Step	What you do in this step	What worksheets you Uuse
Set up ownership and public authority	Establish ownership and public authority for libraries and objects.	Application installation
Create an authorization list	Create authorization lists.	Authorization list
Set up specific authority for objects and libraries	Set up access to libraries and individual objects.	Library description
Secure your printer output queue	Protect printer output by creating output queues and assigning output.	Output queue and workstation security
Secure your workstations	Protect workstations.	Output queue and workstation security

Table 102. Steps in Setting Up Network Security

Step	What you do in this step	What reference you use
Save security information	Save system values, group and user profiles, job descriptions, and resource security information.	Backup and Recovery book
Restore security information	Restore system values, user profiles, objects, authority, programs, authorization lists, and the operating system.	Backup and Recovery book
Set up network security	Set up network security for APPC, and TCP/IP applications.	 Security Planner

Set up your user environment

This topic describes how to set up your user environment and sign on to the system.

To begin setting up user security, you need to set up the overall environment for your users. Use the SETUP menu to set system values, and create your own user profile. You also need to change user IDs and passwords for the Dedicated Service Tools (DST) profiles.

In the following procedures, you will find example command-line screens that illustrate these steps. However, these examples do not show the entire screen. They show only the information necessary to complete the task.

What forms are needed?

Enter information from the system values selection worksheet that you prepared in Plan your security strategy. To set up your overall environment, you need to complete these tasks:

1. "Signing on to the system" on page 172

2. “Selecting the right assistance level”
3. “Preventing others from signing on”
4. “Enter signon system values for security” on page 173
5. “Applying the new system values” on page 175
6. “Creating a security officer profile” on page 175

Signing on to the system

To begin setting up your system environment, you need to sign on to the system.

1. At the console, sign on as the security officer (QSECOFR). If you are signing on for the first time, use the password QSECOFR. Because the system ships this password as expired, the system will prompt you to change this password. You must change this password to successfully sign on.
2. Enter SETUP in the Menu field on the Sign On display.

Note: The SETUP menu is called the Customize Your System, Users, and Devices menu. This text refers to it as the SETUP menu throughout.

```
                    Sign On
                              System . . . . .
                              Subsystem . . . .
                              Display . . . . .
User . . . . . . . . . . . . . QSECOFR
Password . . . . . . . . . . . _________
Program/procedure . . . . . . . _________
Menu . . . . . . . . . . . . . SETUP
Current library . . . . . . . . _________
```

After you sign on to the system, you must select the appropriate assistance level.

Selecting the right assistance level

After signing on to the system, you can choose the appropriate assistance level for users. The assistance level determines what version of a display you see. Many system displays have two different versions:

- A basic assistance level version, which contains less information and does not use technical terminology.
- An intermediate assistance level version, which shows more information and uses technical terms.

Some fields or functions are available only on a particular version of a display. The instructions tell you which version to use. To change from one assistance level to another, use F21 (Select assistance level). F21 is not available from all displays. After you select your assistance level, you must prevent others from signing on to the system while you set up security.

Preventing others from signing on

After you select the right assistance level, you must prevent anyone else from signing on to the system. If you are concerned about people tampering with your system before you have a chance to secure it, you can prevent anyone from signing on at another workstation. This is optional. Do it only if you feel that temporary security is necessary:

1. From the SETUP menu, press F9 to display a command line.
2. On the command line, type GO DEVICESTS.
3. The screen shows the Device Status Tasks menu. If you see the Work with Configuration Status menu, use F21 (Select assistance level) to change to basic assistance level.
4. Select option 1 (Work with display devices).

5. On the Work with Display Devices display, make all the workstations except the one you are using unavailable. Do this by typing 2 in front of each workstation name and pressing the Enter key.
6. Return to the SETUP menu by pressing F3 (Exit) twice.
7. Press F12 (Cancel) to remove the command line.

```
             Work with Display Devices

Type options below, then press Enter.
   1=Make available   2=Make unavailable  5=Display
   7=Display message  8=Work with controller and line
   13=Change description

Opt  Device  Type  Status
__   DSP01   3196  QSECOFR
2_ DSP02 3196 Available to use
2_ DSP03 3196 Available to use
2_ DSP04 3196 Available to use
```

When you make a device unavailable, it does not have a Sign On display, even if it is powered on. Workstations stay unavailable only until you stop and start your system again. You may need to repeat this step.

Enter signon system values for security

After you have prevented others from signing on, you need to enter system values into the system. Use this procedure to enter the information from Part 1 of your System Values Selection form:

1. From the SETUP menu, select option 1 (Change system options).
2. Enter information from your System Values Selection form on the Change System Options display. If you do not want to change one of the choices on the display, you can use the Tab key to skip over it.
3. Enter the correct date and time on this display, if they were not set when you started the system.
4. After you type the information on this page, page down to the next page.
5. Type your choices on the second page of the display and page down.
6. Type your choices on the third page of the display and press the Enter key.
7. You should see the SETUP menu again. Notice the message at the bottom of your display: `System options successfully changed. IPL required.` (The system requires an IPL only if you changed the security level.)

The following table describes possible errors and recovery steps. Use these tables for assistance if your results are different from those described.

Table 103. Possible errors and recovery steps

Possible error	Recovery steps
The MAIN menu is displayed.	You pressed F3 (Exit) or F12 (Cancel). Type `GO SETUP` and try again.
You see another display, such as the Change Cleanup Options display.	You selected the wrong option from the SETUP menu. Press F3 (Exit) to return to the menu and try again.
The Change System Option display is shown again after you press the Enter key.	Look for an error message at the bottom of the display. You probably typed a value that is not allowed. Use F1 (Help) if you need more information. Use F5 (Refresh) if you want the system to restore all the values to what they were before you started typing. Try again.

Table 103. Possible errors and recovery steps (continued)

Possible error	Recovery steps
You pressed the Enter key before you typed all your choices on the display.	You can use this display as many times as necessary to change system values. Select option 1 from the SETUP menu and enter the values you missed the first time. **Attention:** Once your system is operational, do not change the security level without consulting a programmer. Also, do not change the system name if you are using iSeries Access or communicating with another computer.
You pressed the Enter key instead of paging down.	Select option 1 from the SETUP menu again and page down to display the second page. Type your choices and press the Enter key.

The following table shows several values that you can set to make it more difficult for an unauthorized person to sign on to your system. If you run the CFGSYSSEC command, it sets these system values to the recommended settings.

Table 104. Recommended system value settings

System Value Name	Description	Recommended Setting
QAUTOCFG	Whether the system automatically configures new devices.	0 (No)
QAUTOVRT	The number of virtual device descriptions that the system will automatically create if no device is available for use.	0
QDEVRCYACN	What the system does when a device reconnects after an error.[1]	*DSCMSG
QDSCJOBITV	How long the system waits before ending a disconnected job.	120
QDSPSGNINF	Whether the system displays information about previous sign-on activity when a user signs on.	1 (Yes)
QINACTITV	How long the system waits before taking action when an interactive job is inactive.	60
QINACTMSGQ	What the system does when the QINACTITV time period is reached.	*ENDJOB
QLMTDEVSSN	Whether the system prevents a user from signing on at more than one workstation at the same time.	1 (Yes)
QLMTSECOFR	Whether users with *ALLJOB or *SERVICE special authority can sign on only at specific workstations.	1 (Yes)[2]
QMAXSIGN	Maximum consecutive, incorrect sign-on attempts (user profile or password is incorrect).	3
QMAXSGNACN	What the system does when the QMAXSIGN limit is reached.	3 (Disable both user profile and device)

Table 104. Recommended system value settings (continued)

System Value Name	Description	Recommended Setting
Note: 1. The system can disconnect and reconnect TELNET sessions when the device description for the session is explicitly assigned. 2. If you set the system value to 1 (Yes), you will need to explicitly authorize users with *ALLOBJ or *SERVICE special authority to devices. The simplest way to do this is to give the QSECOFR user profile *CHANGE authority to specific devices.		

After entering your system values, you must then apply the new system values.

For more information, see "Values That Are Set by the Configure System Security Command" in the *iSeries Security Reference*.

Applying the new system values

After you enter your system values, you need to apply some of these values. Most changes to system values take effect immediately. However, when you change the security level on your system, the change does not take effect until you stop your system and start it again. After you verify that you typed all the values on the Change System Options display correctly, you are ready to apply the new values.

Note: Attach your workstations to the system, if you have not already done so. When you start the system, it automatically configures those devices using the naming format you chose on the Change System Options display.

Use the following procedure to stop your system and start it again. When your system starts, the values you entered on the Change System Options display take effect.

1. Make sure you have signed on at the console and that no other workstations are signed on.
2. Make sure that the keylock switch on the processor unit is in the Normal position.
3. From the SETUP menu, select the option for Power On and Off Tasks.
4. Select the option to power off the system immediately and then power on. Press the Enter key.
5. The system shows a display that requests you to confirm your power-down request. Press F16 (Confirm).

This causes the system to stop and then start again automatically. Your display goes blank for a few minutes. Then you should see the Sign On display again.

After you apply your new system values, you must create a security officer profile for yourself on the system.

Creating a security officer profile

A security officer on the system is any user with *SECOFR user class or *ALLOBJ and *SECADM special authorities.

After you apply the system values from the Change System Option display, create a user profile for yourself and for the alternate security officer. In the future, use your profile, rather than the QSECOFR profile, when you perform security officer functions.

1. Sign on to the system as QSECOFR and request the SETUP menu. Notice that the system name you chose appears in the upper right of the Sign On display.

2. From the SETUP menu, select the Work with user enrollment option. The Work with User Enrollment display lists the profiles currently on your system. (If you see the Work with User Profile display, press F21 (Select assistance level) and change to basic assistance level.)
3. To create a new profile, type 1 (Add) in the Opt (option) column and the name of your profile in the User column. Press the Enter key.
4. On the Add User display, assign yourself a password.
5. Fill in the fields shown on the sample display with your own appropriate information.
6. Page down to the next page of the display.
7. Fill in the second page of the display and press the Enter key.
8. Check for confirmation messages at the bottom of the Work with User Enrollment display.
9. Press F3 (Exit) to return to the SETUP menu.

After you create a security officer profile for yourself, you need to change user ID and passwords for Service Tools users.

Change known passwords

To keep your system secure, change known passwords for user profiles and dedicated service tools.

Do the following to close some well-known entrances into the server that may exist on your system.

1. Make sure that no user profiles still have default passwords (equal to the user profile name). You can use the Analyze Default Passwords (ANZDFTPWD) command.
2. Try to sign on to your system with the combinations of user profiles and passwords that are shown in Table 105. These passwords are published, and they are the first choice of anyone who is trying to break into your system. If you can sign on, use the Change User Profile (CHGUSRPRF) command to change the password to the recommended value.
3. Start the Dedicated Service Tools (DST) and try to sign on with the passwords that are shown in Table 106 on page 177.
4. If you can sign on to DST with any of these passwords, you should change the passwords.
5. Make sure that you cannot sign on just by pressing the Enter key at the Sign On display without entering a user ID and password. Try several different displays. If you can sign on without entering information on the Sign On display, do one of the following:
   - Change to security level 40 or 50 (QSECURITY system value). (Your applications might run differently when you increase your security level to 40 or 50.)
   - Change all of the workstation entries for interactive subsystems to point to job descriptions that specify USER(*RQD).

Table 105. Passwords for IBM-supplied profiles

User ID	Password	Recommended value
QSECOFR	QSECOFR[1]	A nontrivial value known only to the security administrator. **Write down the password that you have selected and store it in a safe place.**
QSYSOPR	QSYSOPR	*NONE[2]
QPGMR	QPGMR	*NONE[2]
QUSER	QUSER	*NONE[2], [3]
QSRV	QSRV	*NONE[2]
QSRVBAS	QSRVBAS	*NONE[2]

Table 105. Passwords for IBM-supplied profiles (continued)

User ID	Password	Recommended value
Note: 1. 'The system arrives with the *Set password to expired value* for the QSECOFR set to *YES. The first time that you sign on to a new system, you must change the QSECOFR password. 2. The system needs these user profiles for system functions, but you should not allow users to sign on with these profiles. This password is shipped as *NONE. When you run the CFGSYSSEC command, the system sets these passwords to *NONE. 3. To run iSeries Access for Windows using TCP/IP, the QUSER user profile must be enabled.		

Table 106. Passwords for Dedicated Service Tools

DST Level	User ID[1]	Password	Recommended Value
Basic capability	11111111	11111111	A nontrivial value known only to the security administrator.[2]
Full capability	22222222	22222222[3]	A nontrivial value known only to the security administrator.[2]
Security capability	QSECOFR	QSECOFR[3]	A nontrivial value known only to the security administrator.[2]
Service capability	QSRV	QSRV[3]	A nontrivial value known only to the security administrator.[2]
Note: 1. A user ID is only required for PowerPC AS (RISC) releases of the operating system. 2. If your hardware service representative needs to sign on with this user ID and password, change the password to a new value after the hardware service representative leaves. 3. The service tools user profile will expire as soon as it is used for the first time.			

Important: DST passwords can only be changed by an authenticated device. This is also true for all passwords and corresponding user IDs that are identical. For more information on authenticated devices, see the Operations Console setup information in the iSeries Information Center.

Use system service tools to change passwords

You also can use system service tools (SST) instead of DST to change passwords.

You can manage and create service tools user IDs from system service tools (SST) by selecting option 8 (Work with service tools user IDs) from the main SST display. You no longer need to go into DST to reset passwords, grant or revoke privileges, or create service tools user IDs.

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

Change 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 Profiles

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 Profiles

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

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

Change signon error messages

This topic discusses how to change signon error messages to discourage hackers who are trying to break into a system.

Hackers like to know when they are making progress toward breaking into a system. When an error message on the Sign On display says Password not correct, the hacker can assume that the user ID is correct. You can frustrate the hacker by using the Change Message Description (CHGMSGD) command to change the text for two signon error messages. The table shows the recommended text.

Table 107. Signon error messages

Message ID	Shipped Text	Recommended Text
CPF1107	CPF1107 – Password not correct for user profile.	Signon information is not correct. (Do not include the message ID in the message text.)
CPF1120	CPF1120 – User *xxxxx* does not exist.	Signon information is not correct. (Do not include the message ID in the message text.)

Set up system security

This information guides you through the process of setting up your system level security.

The Security wizard can automatically configure your system to the correct system value settings for your company.

Security system values are used to control security on your system.

There are some system values that you can lock down to prevent even these users from changing these system values during normal operation.

Security Wizard

This wizard can automatically configure your system to the correct system value settings for your company.

If you are unsure about how to properly set security-related system values or want to examine your current security policy, complete the Security wizard. This wizard can automatically configure your system to the correct system value settings for your company. You are provided with many options of how to carry out your configuration.

The following are some options that the wizard allows you to do:

- Automatically configure your system's system values based on the information you provide.
- Save your report so you can configure your system at a later date.
- Print a report that includes the recommended system value settings for your system with the implications of such settings.

To access the Security wizard, complete the following steps:

1. In iSeries Navigator, select your system
2. Right-click Security
3. Select Configure

Then, complete the Security wizard.

Apply security system values

Security system values are used to control security on your system.

These values are broken into four groups:

1. General security system values
2. Other system values related to security
3. System values that control passwords
4. System values that control auditing

Deciding which security system values you should use for your business can be perplexing. If you are new to security implementation on servers, or the environment in which you run your server has recently changed, the Security Wizard can help you with decisions.

After you enter your system values, you need to apply some of these values. Most changes to system values take effect immediately. However, when you change the security level on your system, the change does not take effect until you stop your system and start it again. After you verify that you typed all the values on the Change System Options display correctly, you are ready to apply the new values.

Note: Attach your workstations to the system, if you have not already done so. When you start the system, it automatically configures those devices using the naming format you chose on the Change System Options display.

Use the following procedure to stop your system and start it again. When your system starts, the values you entered on the Change System Options display take effect.

1. Make sure you have signed on at the console and that no other workstations are signed on.
2. Make sure that the keylock switch on the processor unit is in the Normal position.
3. From the SETUP menu, select the option for Power On and Off Tasks.
4. Select the option to power off the system immediately and then power on.
5. Press the Enter key. The system shows a display that requests you to confirm your power-down request.
6. Press F16, Confirm. This causes the system to stop and then start again automatically.

Your display goes blank for a few minutes. Then you should see the Sign On display again. After you apply your new system values, you must create a security officer profile for yourself on the system.

This topic provides these links:

- Values set by the Configure System Security command table.
- System Value Finder

Lockdown system values

There are some system values that you can lock down to prevent even these users from changing these system values during normal operation.

Most security system values can be altered only by a user with Security administrator (*SECADM) and All object (*ALLOBJ) special authorities. To prevent even these users from changing these system values during normal operation, system service tools (SST) and dedicated service tools (DST) provide an option to lock these security values.

See the Lock function of security-related system values topic to see a list of the system values that you can lock down.

Set up user security

Setting up user security involves installing application libraries, and setting up user groups and profiles.

This topic guides you through the tasks necessary to set up user security on your system by using the command line interface. The following table highlights each of the steps involved in setting up user security.

Table 108. Steps in setting up user security

Step	What you do in this step	What worksheets you use
Load applications	Create owner profiles. Load your applications. Application libraries and objects should be on the system before you complete the remaining steps.	System values selection Application description
Set up user groups	Create job descriptions, group libraries, and group profiles.	User group description
Create profiles for users in the group	Create individual user profiles	“User profile worksheet” on page 106
Create a personal library for each member of the group	Create individual libraries	Library description

Related concepts

“User security” on page 12
From a user’s point of view, security affects how they use and complete tasks on the system.

Install application libraries

This topic covers the security steps necessary to load your application libraries to your system.

You should load application libraries to the system before setting up user groups and individual profiles. You need to refer to application objects when you create job descriptions and profiles. If you are not able to load your applications before creating group and individual profiles, you may receive warning messages, such as the following:

- The system does not find initial libraries when you create job descriptions.
- The system does not find the initial program or menu when you create profiles.

You cannot successfully test job descriptions and profiles until you load your application libraries.

To load each of your applications, complete these tasks:

Create an owner profile:

This article describes the steps for creating an owner profile, which is required before you can set up user groups.

Before you can create owner profiles for your applications, you need to sign on to the system.

Signing on to the system

- To create owner profiles:

  Profile
  Your own (*SECADM authority is required)

  Menu MAIN
- To load application libraries:

  Check with your application provider to see if you should be signed on as the security officer or the application owner when you load the application libraries. After you sign on, you can create an owner profile for your applications.

Creating an owner profile

After signing on to the system, check your Application description to see if you need to create any profiles before you load the application. To create a profile:

1. Type CRTUSRPRF (Create User Profile) and press F4 (Prompt).
2. On the Create User Profile display, fill in the fields as instructed by your programmer or application provider.
3. Use F10 (More fields) and page down to display additional fields.
4. Check the bottom of your display for messages.

```
          Create User Profile (CRTUSRPRF)
Type choices, press Enter.

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

After you create an owner for the application, you can begin to load your application.

See Create a group profile for additional information.

Load applications:

You can use Application Administration to load applications.

Application Administration is an optionally-installable component of iSeries Navigator, the graphical user interface (GUI) for the system. Application Administration allows system administrators to control the functions or applications available to users and groups on a specific server. This includes controlling the functions available to users that access their server through clients. It is important to note here, that if you access the server from a Windows client, the server user and not the Windows user determines which functions are available for administration.

After you load your applications, you can "Set up user groups."

The Application administration has additional information on applications.

Set up user groups

This article describes the tasks for setting up user groups.

In this task, you will create group libraries, job descriptions, and group profiles. Work through the entire topic with one of your user groups, then go back and repeat the steps for any additional groups.

Use the User Group Description forms that you prepared in "Plan user groups" on page 97.

The next topics will step you through setting up your user groups.

Create a library for a group:

This article describes how to create a library for the user group. You can use libraries to store objects such as programs.

Before you set up user groups, sign on to the system using your own profile (*SECADM authority is required). Go to the MAIN menu.

After you sign on to the system, you need to create a library for the user group. If you plan to have the group share a library for objects they create, such as Query programs, create the library before you create the group profile:

1. Type CRTLIB (Create Library) and press F4 (Prompt).
2. Fill in the display. The library name should be the group profile name.
3. Press F10 (Additional parameters).
4. Fill in the public authority for the library and new objects that are created in the library.
5. Press the Enter key. Check the confirmation message.

```
             Create Library
Type choices, press Enter.
Library  . . . . . . . . . . . . . DPTWH
Library type . . . . . . . . . . . *PROD
Text 'Description' . . . . . . . . Warehouse Library

Additional Parameters
Authority  . . . . . . . . . . . . *USE
Auxiliary storage pool ID  . . . . 1
Create authority . . . . . . . . . *CHANGE
Create object auditing . . . . . . *SYSVAL
```

Possible Error	Recovery
You pressed the Enter key before you typed a description for the library.	Type CHGLIB and press F4 (Prompt). Type the library name on the prompt display and press the Enter key. Type the description on the Change Library display.
You gave the library the wrong name.	Use the Rename Object (RNMOBJ) command.

Create a job description for a group:

This article describes how to create a job description for the group. 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.

After you create a library for the group, you can create a job description for each group.

If the libraries needed for the initial library list are not yet on the system, you receive a warning message when you create the job description.

1. Type CRTJOBD (Create Job Description) and press F4 (prompt).
2. Fill in these fields:

   Job description:
   Same as group profile name.

   Library name:
   QGPL Text: Group description
3. Press F10 (Additional parameters).
4. Page down to the Initial Library List field.

```
          Create Job Description

Type choices, press Enter.

Job description . . . . . . . . . DPTSM
  Library . . . . . . . . . . . . QGPL
Job queue . . . . . . . . . . . . QBATCH
  Library . . . . . . . . . . . . *LIBL
Job priority (on JOBQ). . . . . . 5
Output priority (on OUTQ) . . . . 5
Print device  . . . . . . . . . . *USRPRF
Output queue  . . . . . . . . . . *USRPRF
  Library . . . . . . . . . . . .
Text 'description'. . . . . . . . Sales and Marketing
```

5. Type a + (plus) over *SYSVAL in the Initial library list field to specify that you want to enter a list of values. Press the Enter key.

```
Accounting code . . . . . . . . . *USRPRF
  .
  .
CL syntax check . . . . . . . . . *NOCHK
Initial library list  . . . . . . +
        + for more values
```

6. In the Initial Library List field, type the names of libraries that are checked off in your user group description worksheet:
   - Put one library name per line.
   - Include QGPL and QTEMP. Every job uses a library called QTEMP to store temporary objects. **All initial library lists must have the QTEMP library.** For most applications, the QGPL library should also be on the initial library list.

- You do not need to include the current (default) library on the library list. The system adds that library automatically at signon.

7. Press the Enter key. Check messages. (Page down to see all messages.)

```
Specify More Values for

Type choices, press Enter.

Initial library list . . . . . . CUSTLIB
                                ITEMLIB
                                COPGMLIB
                                ICPGMLIB
                                QGPL
                                QTEMP
```

Possible Error	Recovery
You pressed the Enter key instead of F10.	To put the correct libraries in the initial library list, type CHGJOBD (Change Job Description) and press F4.
You get error messages when you try to create the job description.	The most common error message occurs when you try to include a library that is not on the system. This message is a warning. The job description is still created with the library in the initial library list. You cannot sign on with a profile that specifies the job description until the library is on the system. If the library is on the system, you might have typed the name incorrectly. Verify the library name and try again.

For more information, see Job Description in Chapter 4 of the *iSeries Security Reference*.

Create a group profile:

This article describes how to create a group profile. You can use a group profile to define authority for a group of users, rather than giving authority to each user individually.

After you create a job description, you can create the group profile using the information from Part 2 of the User Group Description form.

1. Use the Work with User Profiles command. Type WRKUSRPRF *ALL. Initially, the display lists the profiles supplied by IBM.

   Note: If you see the Work with User Enrollment display, press F21 to change to intermediate assistance level.

2. To create a new profile, type 1 in the *Opt* (option) column and profile name in the User Profile column. Press the Enter key.

```
          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    DPTSM
     QDOC     Document User Profile
     QSECOFR  Security Officer User Profile
```

3. Type information from your User Group Description form into the appropriate fields.
4. Use the Tab key to skip over any fields where you want to use the default value.

5. Press F10 (Additional parameters).
6. Page down.

```
             Create User Profile (CRTUSRPRF)

Type choices, press Enter.

User profile  . . . . . . . . > DPTSM
User password . . . . . . . . . *none
Set password to expired . . . . *NO
Status  . . . . . . . . . . . . *ENABLED
User class  . . . . . . . . . . *USER
Assistance level  . . . . . . . *SYSVAL
Current library . . . . . . . . *CRTDFT
Initial program to call . . . . cpsetup
Library   . . . . . . . . . . . cppgmlib
Initial menu  . . . . . . . . . cpmain
Library . . . . . . . . . . . . cppgmlib
Limit capabilities  . . . . . . *yes
Text 'description'. . . . . . . Sales and Marketing
```

7. Enter the remaining fields from your User Group Description form on the additional pages of the display and press the Enter key.

```
             Create User Profile

             Additional Parameters

Special authority . . . . . . . *USRCLS
.
.
Job description . . . . . . . . DPTSM
  Library . . . . . . . . . . . QGPL
```

8. Check messages.

```
             Create User Profile

Group authority . . . . . . . . *NONE
.
.
Print device  . . . . . . . . . PRT03
```

Important: A group profile is just a special type of user profile. Many messages and displays refer to group profiles as users or user profiles. The system only knows that you have created a group profile if you add members to it or assign a group identification number (gid) to it.

Possible Error	Recovery
You pressed the Enter key before typing all the values in the group profile.	Press F5 (Refresh) to add the profile that you created to the Work with User Profiles display. Use Option 2 (Change) to correct the profile.
You created a profile with the wrong name.	You cannot change the name of a profile. Use the Copy Option (3) to create a new profile with the correct name. Then delete (Option 4) the profile with the wrong name.
Some of the fields from the User Group Description form do not appear on the display.	Make sure you are using intermediate assistance level. The basic assistance level version of Create User Profile is called the Add a User display. To change assistance levels, press F12 (Cancel) to return to the Work with User Enrollment display. Use F21 to change assistance levels.

Possible Error	Recovery
You accidentally erased some of the default information from the Create User Profile display.	If you leave a field blank, the system uses the default value when the user profile is created. If you want to see the default values, press F5 (Refresh) to restore the entire display. Type your information again.

Listing your results

List the names and descriptions of all profiles on the system by using the Display Authorized Users (DSPAUTUSR) command. Type `DSPAUTUSR OUTPUT(*PRINT)`. Check to make sure that all group profiles have a password of *NONE.

Complete the following before you set up individual users:

- Create a job description for each user group.
- Optionally, create a library for each group.
- Create a group profile for each user group.

For more information about group profiles and IBM-supplied user profiles, see the following topics in the *iSeries Security Reference*:

- Planning Group Profiles in Chapter 7
- IBM-Supplied User Profiles in Chapter 9

Related concepts

"Group profiles" on page 8
Group profiles define authority for a group of users.

Create profiles for users in the group:

This topic describes how to create profiles for individual users.

When you set up user groups, you completed the steps to create group profiles. Now, you create individual profiles for the members of the groups. Work through the entire topic with the members of one user group, then go back and repeat the steps for any additional groups.

Use the Individual User Profile worksheet that you prepared in "Plan user profiles" on page 104.

To create individual profiles for the members of the groups, complete these tasks:

1. Create a personal library (optional).
2. Copy the group profile.
3. Set the password to expire.
4. Create additional users (optional).
5. Change information about a user, if necessary.
6. Display your results.

Note: Repeat creating a personal library and creating additional users, until every group member has a user profile.

For more information, see Job Description in Chapter 4 of the *iSeries Security Reference*.

Related concepts

"Plan user profiles" on page 104
This topic describes the purpose of user profiles and how to design them.

Create a personal library for each member of a group:

This article describes the task, create a personal library for each member of the group, explains why it is important, and provides step-by-step instructions.

To begin setting up individual users, you may need to create a personal library for each member for objects, such as Query programs. Create personal libraries before you create the individual user profiles.

1. Type CRTLIB and press F4 (Prompt).
2. Give the library the same name as the user profile.
3. Press F10 (Additional parameters).
4. Fill in the public authority for the library and new objects that are created in the library.
5. Press the Enter key. Check the confirmation message.

```
              Create Library

Type choices, press Enter.

Library  . . . . . . . . . . . . DPTSM
Library type . . . . . . . . . . *PROD
Text 'description' . . . . . . . Warehouse Library

             Additional Parameters

Authority  . . . . . . . . . . . *EXCLUDE
Auxiliary storage pool ID  . . . 1
Create authority . . . . . . . . *CHANGE
Create object auditing . . . . . *SYSVAL
```

After you create a personal library, you can create the individual profile by copying the group profile.

For more information on libraries, see the following sections in the iSeries Security Reference:

- "Security Risks of Library Lists" in Chapter 6
- "Planning Libraries" in Chapter 7

Copy the group profile:

This article describes how to copy the group profile, explains why it is important, and provides step-by-step instructions.

The group profile has two roles:

1. The system uses it to determine whether a group member is authorized to use an object.
2. You can use it as a pattern to create user profiles for the individual group members.

When you set up user groups, you created group profiles. Now, you can copy a group profile to create an individual profile and copy the individual profile to create other profiles in the group.

1. Select the Work with User Enrollment option from the SETUP menu.

   Tip: If you see the Work with User Profiles display, use F21 (Select assistance level) to change to basic assistance level.
2. Type 3 (Copy) in the *Opt* column in front of the user group. The screen shows the Copy User display. (If the user group you want to copy is not on your display, page down until you find it.) The system leaves the user name field blank and fills in the remaining fields from the group profile that you copied.

```
             Work with User Enrollment

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

Opt   User    Description
      DPTSM   Sales and Marketing Department
3     DPTWH   Warehouse Department
```

3. Type the name and description of the user profile that you are creating.
4. Leave the password blank. The system automatically makes the password the same as the new user profile name.
5. Put the group profile name in the User Group field.
6. Check your Individual User Profile worksheet to see if this user has other values that are different from the group. Enter those values.
7. Page down.

```
             Copy User

Copy from user . . . . :  DPTWH

Type choices below, then press Enter.

User . . . . . . . . . . WILLISR
User description . . . . Willis, Rose
Password . . . . . . . .
Type of user . . . . . . *SYSOPR
User group . . . . . . . DPTWH

Restrict command line use N

Default library  . . . . DPTWH
Default printer  . . . . PRT04
Sign on program  . . . . *NONE
Library  . . . . . . . .
First menu . . . . . . . ICMAIN
  Library  . . . . . . . ICPGMLIB
```

8. Make any changes that are necessary on the next page of the display and press the Enter key.
9. Check for confirmation messages at the bottom of the Work with User Enrollment display.

```
             Copy User

Copy from user . . . . :  DPTWH

Type choices below, then press Enter.

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

Possible Error	Recovery
You see the Create User Profile display instead of the Copy User display.	Use F12 (Cancel) to return to the Work with User Profiles display. Use F21 to change to basic assistance level. Start the copy operation again.
The user profile name that you have selected will not fit in the user prompt.	Although user profile names may be up to 10 characters, the Copy User and Add User displays support no more than 8 character names. Either choose a shorter user name or use the intermediate assistance level to create individual user profiles.

Testing the User Profile

When you create the first individual profile in a group, you should test it by signing on with that profile. Verify that you see the correct first menu and that the signon program runs.

If you are unable to sign on successfully with the profile, the system probably could not find something specified in the profile. This could be the signon program, the job description, or one of the libraries in the initial library list. Use the Work with Printer Output display to find the job log that was written when you tried to sign on. The job log tells you what errors occurred.

After you test the user profile, you can set the password to expire.

Using an Individual Profile as a Group Profile

Creating profiles specifically to be group profiles is preferable to making existing profiles into group profiles. You may find that a specific user has all the authorities needed by a group of users and be tempted to make that user profile into a group profile. However, using an individual's profile as a group profile may cause problems in the future:

- If the user whose profile is used as the group profile changes responsibilities, a new profile needs to be designated as the group profile, authorities need to be changed, and object ownership needs to be transferred.
- All members of the group automatically have authority to any objects created by the group profile. The user whose profile is the group profile loses the ability to have private objects, unless that user specifically excludes other users.

Try to plan group profiles in advance. Create specific group profiles with password *NONE. If you discover after an application has been running that a user has authorities that should belong to a group of users, do the following:

1. Create a group profile.
2. Use the GRTUSRAUT command to give the user's authorities to the group profile.
3. Remove the private authorities from the user, because they are no longer needed. Use the RVKOBJAUT or EDTOBJAUT command.

Set the group profile password to expire:

This article describes how to set the group profile password to expire, explains why it is important, and provides step-by-step instructions.

Set up individual profiles to require that users change their passwords the first time they sign on. The Set Password to Expire field does not appear on the basic assistance level version of the Copy User display. You need to change it separately, after you create the user profile with the copy function. To change the Set Password to Expire field, type `CHGUSRPRF profile-name PWDEXP(*YES)`.

Note: If you want to test a user profile by signing on with it, do the test before you set the password to expire.

Possible Error	Recovery
You tested a profile and were forced to change the password.	Type `CHGUSRPRF` *profile-name* and press F4 (Prompt). Set the password back to the user profile name. (Type the user profile name in the password field.) Type `*YES` in the Set Password to Expire field. You need intermediate assistance level to do this.

After you create the first individual user profile, you can create additional users.

For more information, see “Set Password to Expired” in Chapter 4 of the iSeries Security Reference.

Create profiles for users not in a group

Copy the first individual user profile to create additional members in the group. Look at each individual profile carefully when you create it with the copy method.

Check your Individual User Profile form and make sure that you change any fields that are unique for the new user profile.

1. On the Work with User Enrollment display, type 3 (Copy) in front of the user profile you want to copy.
2. On the Copy User display, type the profile name and description.
3. Enter information into any fields that are unique for the new user.

```
             Work with User Enrollment

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

Opt   User    Description
      DPTSM   Sales and Marketing Department
      DPTWH   Warehouse Department
3     WILLISR Willis, Rose
```

Possible Error	Recovery
The profile that you want to copy does not appear on the Work with User Enrollment display.	Press F5 (Refresh). Page up and page down. The list is alphabetical by profile name.

Changing information about a user

For some users, you may need to set values that do not appear on the Copy User display. For example, some users may belong to more than one group profile. After you have created a user profile by using the copy method, you can change it.

1. On the Work with User Enrollment display, press F21 to change to intermediate assistance level.
2. On the Work with User Profiles display, type a 2 (Change) in the *Opt* (option) column next to the profile you want to change. Press the Enter key.

```
             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
2    AMESJ    Ames, Janice
     DPTSM    Sales and Marketing Department
     QDOC     Document User Profile
     QSECOFR  Security Officer User Profile
     WAGNERR  Wagner, Ray
     WILLISR  Willis, Rose
```

3. On the Change User Profile display, press F10 (Additional parameters).
4. Page down until you find the fields that you want to change. For example, if you want to make the user a member of additional group profiles, page down until you find the Supplemental Groups field.
5. Type the values you want and press the Enter key. You receive confirmation messages and see the Work with User Profiles display again.

```
             Change User Profile (CHGUSRPRF)

Type choices, press Enter.

Maximum allowed storage . . . . *NOMAX
Highest schedule priority . . . 3
Job description . . . . . . . . DPTWH
Library . . . . . . . . . . . . QGPL
Group profile . . . . . . . . . DPTWH
Owner . . . . . . . . . . . . . *GRPPRF
Group authority . . . . . . . . *USEE
Group authority type  . . . . . *PGP
Supplemental groups . . . . . . DPTIC
          + for more values
```

Once you have changed the user information, you can display your results to check your profiles.

Displaying user profiles

Several methods are available to display the profiles that you created.

Displaying one profile
Use option 5 (Display) from either the Work with User Enrollment display or the Work with User Profiles display.

Listing one profile
Use the Display User Profile command: DSPUSRPRF *profile-name* DETAIL(*BASIC) OUTPUT(*PRINT).

Displaying group members
Type DSPUSRPRF *group-profile-name* *GRPMBR. You can use OUTPUT(*PRINT) to print the list.

Listing all profiles
To list the names and descriptions of all profiles, sorted by group, use the Display Authorized Users command: DSPAUTUSR SEQ(*GRPPRF) OUTPUT(*PRINT).

Before you set up ownership and public authority, complete these tasks:

- Finish creating all your individual user profiles.
- Set the password to expire for each profile.
- Print a list of all profiles sorted by group and keep it with your User Group Description forms. Print the list again when you add new users.

Related concepts

"Plan user profiles" on page 104
This topic describes the purpose of user profiles and how to design them.

Limit access to program functions

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.

The limit access to program function allows you to provide security for the program when you do not have an object to secure for the program. There are two methods that you can use to manage user access to application functions through iSeries Navigator.

The first method uses Application Administration:

1. Right-click the system that contains the function whose access setting you want to change.
2. Select **Application Administration**.
3. If you are on an administration system, select **Local Settings**. Otherwise, continue with the next step.

4. Select an administrable function.
5. Select **Default Access** to allow all users to access the function by default.
6. Select **All Object Access** to allow all users with all object system privilege to access this function.
7. Select **Customize**and use the **Add** and **Remove** buttons on the **Customize Access** dialog to add or remove users or groups in the **Access Allowed** and **Access Denied** lists.
8. Select **Remove Customization** to delete any customized access for the selected function.
9. Click **OK** to close the **Application Administration** dialog.

The second method of managing user access uses iSeries Navigator's Users and Groups:

1. In iSeries Navigator, expand **Users and Groups**.
2. Select **All Users**, **Groups**, or **Users Not in a Group** to display a list of users and groups.
3. Right-click a user or group, and select **Properties**.
4. Click **Capabilities**.
5. Click the **Applications** tab.
6. Use this page to change the access setting for a user or group.
7. Click **OK** twice to close the **Properties** dialog.

Important: The limit access to program function does not prevent a user from accessing a resource, such as a file or program, from another interface. You still need to use resource security.

The limit access to program function support provides APIs to:

- Register a function
- Retrieve information about the function
- Define who can or cannot use the function
- Check to see if the user is allowed to use the function

To use this 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, the application calls the check usage API to see if the user is allowed to use the function that is associated with the code block, before invoking the code block. If the user is allowed to use the registered function, the code block is run. If the user is not allowed to use the function, the user is prevented from running the code block.

The system administrator specifies who is allowed or denied access to a function. The administrator can either use the Work with Function Usage Information (WRKFCNUSG) command to manage the access to program function, or the iSeries Navigator.

Implement resource security

This information helps you establish resource security for workstations and printers by setting ownership and public authority to objects, as well as specific authority to applications.

Your most important protection is resource security on your server. Resource security on the system allows you to define who can use objects and how those objects can be used. The ability to access an object is called authority. When you set up object authority, you can need to be careful to give your users enough authority to do their work without giving them the authority to browse and change the system. Object authority gives permissions to the user for a specific object and can specify what the user is allowed to do with the object. An object resource can be limited through specific detailed user authorities, such as adding records or changing records.

System resources can be used to give the user access to specific system-defined subsets of authorities: *ALL, *CHANGE, *USE, and *EXCLUDE. Files, programs, libraries, and directories are the most common system objects that require resource security protection, but you can specify authority for any individual object on the system.

Defining Who Can Access Information

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

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

You define who can use an object in several ways:

Public Authority
The public consists of anyone who is authorized to sign on to your system. Public authority is defined for every object on the system, although the public authority for an object may be *EXCLUDE. Public authority to an object is used if no other specific authority is found for the object.

Private Authority
You can define specific authority to use (or not use) an object. You can grant authority to an individual user profile or to a group profile. An object has private authority if any authority other than public authority, object ownership, or primary group authority is defined for the object.

User Authority
Individual user profiles may be given authority to use objects on the system. This is one type of private authority.

Group Authority
Group profiles may be given authority to use objects on the system. A member of the group gets the group's authority unless an authority is specifically defined for that user. Group authority is also considered private authority.

Object Ownership
Every object on the system has an owner. The owner has *ALL authority to the object by default. However, the owner's authority to the object can be changed or removed. The owner's authority to the object is not considered private authority.

Primary Group Authority
You can specify a primary group for an object and the authority the primary group has to the object. Primary group authority is stored with the object and may provide better performance than private authority granted to a group profile. Only a user profile with a group identification number (gid) may be the primary group for an object. Primary group authority is not considered private authority.

Defining How Information Can Be Accessed

Authority means the type of access allowed to an object. Different operations require different types of authority.

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

Authority to an object is divided into three categories:

1. Object Authority defines what operations can be performed on the object as a whole.
2. Data Authority defines what operations can be performed on the contents of the object.
3. Field Authority defines what operations can be performed on the data fields.

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

Many 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. Special authority is required by some CL commands for objects and the object libraries. Using library security is one technique for protecting information while maintaining a simple security scheme.

Although library security is a simple, effective method for protecting information, it may not be adequate for data with high security requirements. Many objects reside in directories. Highly sensitive objects should be secured individually or with an authorization list, rather than relying on library security.

You will need the following worksheets during this process:

- The Application Installation worksheet, prepared in ″Planning your application installation.″
- The Authorization List worksheet, prepared in ″Grouping objects.″
- The Library Description worksheet, prepared in ″Determining ownership of libraries and objects.″
- The Output Queue and Workstation Security worksheet, prepared in ″Protecting printer output″ and ″Protecting workstations.″
- The System Responsibilities worksheet, prepared in ″Planning your overall security strategy.″

Complete the following tasks:

- Set up ownership and public authority
- Create authorization lists
- Secure objects with an authorization list
- Add users to the authorization lists
- Set up any specific authorities
- Secure workstations
- Secure printer output
- Restrict access to the system operator message queue

Related concepts

“Resource security” on page 14
You can use resource security on the system to control the actions of authorized users after successful authentication.

Set up ownership and public authority

In this topic, you establish ownership and public authority for application libraries, group libraries, and personal libraries.

You should work through this process with one application, and then go back and repeat the steps for any additional applications. The sample displays show the Application Installation forms that Sharon Jones prepared for the Customer Orders application in ″Planning your application installation.″

Use the procedures in this topic whenever you install a new application on your system or when you set up security for an existing application. Use the Application Installation forms that you prepared in ″Planning your application installation.″

In order to set up ownership and public authority, complete these tasks:

1. Create the owner profile

2. Change library ownership
3. Set ownership of application objects
4. Set public access to a library
5. Set public authority for all objects in a library
6. Set public authority for new objects
7. Work with group and personal libraries

Sign on to the system.

Profile
Your own (*ALLOBJ authority is required)

Menu MAIN

Create an owner profile:

This topic outlines the process of creating an owner profile.

If the owner profile does not yet exist, use the CRTUSRPRF (Create User Profile) command to create it. Set the password to *NONE.

If the owner profile already exists, use the CHGUSRPRF (Change User Profile) command to set the password to *NONE.

After you create the owner profile, you can Change library ownership

Change library ownership:

This step changes the ownership of a library, not the objects in the library.

Attention: Be sure to check with your application provider before you change ownership of any application objects. Some applications use functions that rely on specific object ownership.

1. Type CHGOBJOWN (Change Object Owner) and press F4 (Prompt).
2. Fill in the library name, object type (*LIB), and new owner.
3. Check confirmation messages.

After you change library ownership, you can set ownership for application objects.

Set up ownership of application objects:

Changing the ownership of application objects is a cumbersome task, because you must change each object individually. If possible, ask your programmer or application provider to establish ownership for you.

Listing the objects in a library

Before you change ownership, print a list of all the objects in the library, using the Display Library command. You can use it as a checklist. Type DSPLIB library-name *PRINT.

Choosing the best method

Choose one of these two methods to change ownership of objects in your application libraries:

Method	What it does	When to use it
The Works with Objects by Owner command	Shows a display which lists all the objects that a profile owns. You use an option on the display to change the owner of an object.	This method is easier to use. However, if either QPGMR or QSECOFR own the objects, IBM does not recommend this method. Those profiles own many objects, and your list display would be very large.
The Change Object Ownership command	Requires using a separate command for each object. However, you can use Retrieve (F9) to repeat the previous command and reduce the amount of typing required.	This method is faster if either QPGMR or QSECOFR own the objects.

Set up public access to a library:

After you set ownership of application objects, you can use the Edit Object Authority (EDTOBJAUT) command to change public authority to the library:

These steps will set up public access to a library on your system.

1. Type EDTOBJAUT library-name *LIB.
2. Move the cursor down to the line showing *PUBLIC.
3. Type the authority which you want the public to have to the library and press the Enter key. The display shows the new authority.

Set up public authority for objects in a library:

Use the Grant Object Authority (GRTOBJAUT) command to set public authority for all the objects in a library.

Note: Use the Revoke Object Authority (RVKOBJAUT) command to remove the current public authority for objects in a library.

1. Type RVKOBJAUT and press F4 (Prompt).
2. Fill in the display as shown, substituting the name of your application library, and press the Enter key.

   Note: If the library has a large number of objects, the system may take a few minutes to process your request.
3. Type GRTOBJAUT and press F4 (Prompt).
4. Fill in the display as shown, substituting the name of your application library and the authority you want, and press the Enter key.

   Note: If the library has a large number of objects, the system may take a few minutes to process your request.

After you have completed setting public authority for all objects in a library, you can use the job log to check your work next.

Set up public authority for new objects:

The library description has a parameter called create authority (CRTAUT), which determines the public authority for new objects that are created in the library. The commands that create objects use the

CRTAUT authority of the object library as the default. You should make the CRTAUT for a library the same as the public authority for the majority of existing objects in the library.

1. Type CHGLIB library-name and press F4 (Prompt).
2. Press F10 (Additional parameters).
3. Enter your choice in the Create authority field.

If you set the CRTAUT to *SYSVAL, the system uses the current setting for the QCRTAUT system value when you create a new object in the library. Setting a specific CRTAUT authority for each library protects against future changes to the QCRTAUT system value.

Work with group and personal libraries:

Your profile owns the group and personal libraries you created when you set up user groups and individual users.

Use the procedures just covered to change ownership of group libraries to the group profile and change ownership of personal libraries to the individual user profiles.

Set the Create Authority parameter for each group and personal library to determine the public authority for any new objects in those libraries.

Before you start creating authorization lists, complete these tasks:

1. Use your Application Installation forms and your Library Description forms to make sure that you have established ownership and public authority for all your application libraries.
2. Set ownership and create authority for all of the group and personal libraries that you created.

Note: You can get a list of all the libraries on your system by typing `DSPOBJD *ALL *LIB *PRINT`.

Create an authorization list

This article describes the task, create an authorization list, explains why it is important, and provides step-by-step instructions.

After you set up ownership and public authority, you are ready to set up authorization lists. Using information from your Authorization List forms, create any authorization lists that are necessary to secure the library.

Use the Create Authorization List (CRTAUTL) command:

1. Type CRTAUTL and press F4 (Prompt).
2. Fill in the information from your Authorization List form.
3. Press F10 (Additional parameters).
4. Use the authority parameter to specify the public authority for objects that are secured by the list.
5. Check for confirmation messages.

Possible error	Recovery
You typed the name of the list incorrectly.	You cannot change the name of a list, once the system has created it. Delete the list (DLTAUTL) and try again.
You forgot to specify the public authority for the list.	Use the Edit Authorization List (EDTAUTL) command.

To use this function, perform the following steps:

1. From iSeries Navigator, expand your server Security. You will see Authorization Lists and Policies.
2. Right-click Authorization Lists and select New Authorization List. The New Authorization List allows you to do the following:

- Use: Allows access to the object attributes and use of the object. The public may view, but not change the objects.
- Change: Allows the contents of the object, with some exceptions, to be changed.
- All: Allows all operations on the object, except those that are limited to the owner. The user or group can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user or group can also change ownership of the object.
- Exclude: All operations on the object are prohibited. No access or operations are allowed to the object for the users and groups having this permission. Specifies the public is not allowed to use the object.

When working with authorization lists you will want to grant permissions for both objects and data.

Object permissions you can choose are:

- Operational: Provides the permission to look at the description of an object and use the object as determined by the data permission that the user or group has to the object.
- Management: Provides the permission to specify the security for the object, move or rename the object, and add members to the database files.
- Existence: Provides the permission to control the object's existence and ownership. The user or group can delete the object, free storage of the object, perform save and restore operations for the object, and transfer ownership of the object. If a user or group has special save permission, the user or group does not need object existence permission.
- Alter (used only for database files and SQL packages): Provides the permission needed to alter the attributes of an object. If the user or group has this permission on a database file, the user or group can add and remove triggers, add and remove referential and unique constraints, and change the attributes of the database file. If the user or group has this permission on an SQL package, the user or group can change the attributes of the SQL package. This permission is currently used only for database files and SQL packages.
- Reference (used only for database files and SQL packages): Provides the permission needed to reference an object from another object such that operations on that object may be restricted by the other object. If the user or group has this permission on a physical file, the user or group can add referential constraints in which the physical file is the parent. This permission is currently used only for database files.

Data permissions you can choose are:

- Read: Provides the permission needed to get and display the contents of the object, such as viewing records in a file.
- Add: Provides the permission to add entries to an object, such as adding messages to a message queue or adding records to a file.
- Update: Provides the permission to change the entries in an object, such as changing records in a file.
- Delete: Provides the permission to remove entries from an object, such as removing messages from a message queue or deleting records from a file.
- Execute: Provides the permission needed to run a program, service program or SQL package. The user can also locate an object in a library or directory.

You can now secure objects with an authorization list.

Related concepts

"Authorization lists" on page 9
Like a group profile, an authorization list allows you to group objects with similar security requirements and associate the group with a list of users and user authorities.

Secure objects with an authorization list:

Once you create an authorization list, use the Edit Object Authority (EDTOBJAUT) command to secure the items listed on your Authorization List form:

Steps required to create an authorization list:

1. Type EDTOBJAUT and press F4 (prompt).
2. Fill in the prompt display and press the Enter key.
3. On the Edit Object Authority display, enter the authorization list name.
4. If the public authority for the object comes from the authorization list, change the public authority to *AUTL. Repeat these steps for each object on your Authorization List form.

You can now add users to the authorization list.

Add users to an authorization list:

Once you secure objects with an authorization list, use the Edit Authorization List (EDTAUTL) command to add the users listed on your Authorization list form:

1. Type EDTAUTL authorization-list-name.
2. On the Edit Authorization list display, press F6 (Add new users).
3. Enter the names of the users or groups and the authority they should have to the items on the list and press the Enter key. The new users should appear on the list.

Possible error	Recovery
You gave a user or group the wrong authority to the list.	You can change the authority on the Edit Authorization List display.
You added the wrong user or group to the list.	You can remove a user or group using the Remove Authorization List Entry (RMVAUTLE) command, or you can type blanks over the user's authority on the Edit Authorization List display.

Checking your work

- Use the Display Authorization List (DSPAUTL) command to list all the user authorities to the authorization list.
- Use F15 from the display to list all the objects secured by the authorization list.

Before you set up specific authorities, complete these tasks:

1. Use the CRTAUTL command to create any authorization lists you need for the application.
2. Secure objects with authorization lists by using the EDTOBJAUT command.
3. Add users to authorization lists by using the EDTAUTL command.

Set up specific authority for objects and libraries

You can use the Edit Object Authority (EDTOBJAUT) command to set specific authority for the library and objects in the library.

In Setting up ownership and public authority, you learned how to use the GRTOBJAUT command to set public authority for all the objects in a library, based on the information on your Library description form. Now you will use EDTOBJAUT, and the information on your Library description form to set specific object and library authorities.

Related concepts

"System-defined authorities" on page 19
This table shows how system-defined authorities apply to securing files, programs, and libraries.

Set up authority for a library:

A library is really a special type of object. You set authority for a library just like you set authority for any other object, by using the EDTOBJAUT command. All libraries reside in the IBM-supplied library that is called QSYS.

Use the Edit Object Authority (EDTOBJAUT) command to set specific authority for the library and objects in the library, based on the information on your Library description worksheet.

1. Type EDTOBJAUT and press F4 (Prompt).
2. Fill in the prompt display and press the Enter key.
3. On the Edit Object Authority display, press F6 (Add new users) to give authority to users whom the display does not list.
4. Press the Enter key.
5. The Edit Object Authority display should match the information on both Parts 1 and 2 of the Library Description form.

The public authority for new objects (CRTAUT) authority does not appear on the Edit Object Authority display for a library. Use the Display Library (DSPLIB) command to see the CRTAUT for a library. You can also use this procedure to set up specific authority to an object on the system. You can now set specific authority for an object.

Set up authority for an object:

The procedure for setting specific authority for an object in an application library is the same as setting specific authority for a library.

1. Type EDTOBJAUT and press F4 (Prompt).
2. Fill in the information on the prompt display and press the Enter key.
3. Fill in the authority information on the Edit Object Authority display and press the Enter key.

You can now set authority for more than one object at a time. Set up authority for multiple objects

Set up authority for multiple objects:

Use the Grant Authority (GRTOBJAUT) command to set security for multiple objects.

Type GRTOBJAUT and press F4 (Prompt).

Note: Many commands allow you to specify the first characters followed by an asterisk (*) for a parameter. The system performs the operation on every object whose name starts with those characters.

Use the DSPJOBLOG command to check your work to verify that the system made the requested authority changes.

Before going to Securing printer output, use the EDTOBJAUT or the GRTOBJAUT command to set up the specific authorities on your Library Description form.

Enforce object authority:

Whenever you try to access an object, the operating system checks your authority to that object.

If the security level on your system (QSECURITY system value) is set to 10 or 20, every user automatically has authority to access every object because every user profile has *ALLOBJ special authority.

Object authority tip: If you are not sure whether you are using object security, check the QSECURITY (security level) system value. If QSECURITY is 10 or 20, you are not using object security.

You must plan and prepare before you change to security level 30 or higher. Otherwise, your users may not be able to access the information that they need.

Set up menu security

This article discusses the user profile parameters for setting up menu security.

The server provides several user profile parameters that you can use to implement menu access control:

- Use **Initial menu** (INLMNU) parameter to control what menu the user first sees after the user signs on.
- Use **Initial program** (INLPGM) parameter to run a setup program before the user sees a menu, or you can use this parameter to restrict a user to running a single program.
- Use **Limit capabilities** (LMTCPB) parameter to restrict a user to a limited set of commands. This parameter also prevents the user from specifying a different initial program or menu on the Sign On display. The LMTCPB parameter only limits commands that are entered from the command line.

For more information on these user profile parameters, see "Initial menu," "Initial program," and "Limit Capabilities" in the iSeries Security Reference.

Related concepts

"Menu security" on page 11
Menu security controls which menu functions a user can perform.

Limitations of menu access control:

You cannot rely solely on menu access control to protect your system and allow users to use the system effectively to do their jobs.

There are a number of limitations to menu access control. Computers and users have changed a great deal in the past few years. Many tools, such as query programs and spreadsheets, are available so that users can do some of their own programming, which lightens the work load of IS departments. Some tools, such as SQL or ODBC, provide the capability to view information and to change information. To enable these tools within a menu structure is very difficult.

As a security administrator who is trying to enforce menu access control, you have two basic problems:

- If you are successful in limiting users to menus, your users will probably be unhappy because their ability to use modern tools is limited.
- If you are not successful, you could jeopardize critical confidential information that menu access control is supposed to protect. When your system participates in a network, your ability to enforce menu access control decreases. For example, the LMTCPB parameter applies only to commands that are entered from a command line in an interactive session. The LMTCPB parameter has no affect on requests from communications sessions, such as PC file transfer, FTP, or remote commands.

Enhance menu access control with object security:

This article provides suggestions for moving toward an object security environment to complement your menu access control.

With the many options that are available to connect to systems, a viable server security scheme for the future cannot rely solely on menu access control. You can enhance menu access control by granting appropriate authority that users must have to objects to run their applications. You then assign users to groups and give the groups appropriate authority. This approach is reasonable and logical. However, if

your system has been operational for many years and has many applications, the task of analyzing applications and setting up object authority probably seems overwhelming.

The solution to this problem could be to use your current menus to set up a transition environment while you gradually analyze your applications and objects.

Tip: Your current menus combined with programs that adopt the authority of the program owners may provide a transition beyond menu access control. Be sure to protect both the programs that adopt authority and the user profiles that own them.

Example: Change the menu control environment:

In this example, you are changing the menu control environment for the Order Entry (OEMENU) menu and the associated files and programs.

This example starts with the following assumptions and requirements:

- All of the files are in the library ORDERLIB.
- You do not know the names of all the files. You also do not know what authority the menu options require to different files.
- The menu and all the programs that it calls are in a library called ORDERPGM.
- You want everyone who can sign on to your system to be able to view information in all the order files, customer files, and item files (with queries or spreadsheets, for example).
- Only users whose current signon menu is the OEMENU should be able to change the files. They must use the programs on the menu to do this.
- System users other than the security administrators do not have *ALLOBJ or *SECADM special authority.

Perform the following steps to change this menu-access-control environment to accommodate the need for queries:

1. Make a list of the users whose initial menu is the OEMENU. You can use the Print User Profile (PRTUSRPRF *ENVINFO) command to list the environment for every user profile on your system. The report includes the initial menu, initial program, and current library.
2. Make sure that the OEMENU object (it may be a *PGM object or a *MENU object) is owned by a user profile that is not used for signon. The user profile should be disabled or have a password of *NONE. For this example, assume that OEOWNER owns the OEMENU program object.
3. Make sure that the user profile that owns the OEMENU program object is not a group profile. You can use the following command: `DSPUSRPRF USRPRF(OEOWNER) TYPE(*GRPMBR)`
4. Change the OEMENU program to adopt the authority of the OEOWNER user profile. Use the CHGPGM command to change the USRPRF parameter to *OWNER. *MENU objects cannot adopt authority. If OEMENU is a *MENU object, you can adapt this example by doing one of the following:
   - Create a program to display the menu.
   - Use adopted authority for the programs that run when the user selects options from the OEMENU menu.
5. Set the public authority to all of the files in ORDERLIB to *USE by typing the following two commands:`RVKOBJAUT OBJ(ORDERLIB/*ALL) OBJTYPE(*FILE) USER(*PUBLIC) AUT(*ALL)GRTOBJAUT OBJ(ORDERLIB/*ALL) OBJTYPE(*FILE) USER(*PUBLIC) AUT(*USE)` Remember that if you select *USE authority, users can copy the file by using PC file transfer or FTP.
6. Give the profile that owns the menu program *ALL authority to the files by typing the following: `GRTOBJAUT OBJ(ORDERLIB/*ALL) OBJTYPE(*FILE) USER(OEOWNER) AUT(*ALL)` For most applications, *CHANGE authority to files is sufficient. However, your applications may perform functions, such as clearing physical file members, that require more authority than *CHANGE. Eventually, you should analyze your applications and provide only the minimum authority that is necessary for the

application. However, during the transition period, by adopting *ALL authority, you avoid application failures that may be caused by insufficient authority.

7. Restrict authority to the programs in the order library by typing: `GRTOBJAUT OBJ(ORDERPGM/*ALL) OBJTYPE(*PGM) USER(*PUBLIC) AUT(*EXCLUDE)`
8. Give the OEOWNER profile authority to the programs in the library by typing: `GRTOBJAUT OBJ(ORDERPGM/*ALL) OBJTYPE(*PGM) USER(OEOWNER) AUT(*USE)`
9. Give the users that you identified in step 1 authority to the menu program by typing the following for each user: `GRTOBJAUT OBJ(ORDERPGM/OEMENU) OBJTYPE(*PGM) USER(user-profile-name) AUT(*USE)`

When you have completed these steps, all system users who are not explicitly excluded will be able to access (but not change) the files in the ORDERLIB library. Users who have authority to the OEMENU program will be able to use the programs that are on the menu to update files in the ORDERLIB library. Only users who have authority to the OEMENU program will now be able to change the files in this library. A combination of object security and menu access control protects the files.

When you complete similar steps for all the libraries that contain user data, you have created a simple scheme for controlling database updates. This method prevents system users from updating database files except when they use the approved menus and programs. At the same time, you have made database files available for viewing, analyzing, and copying by users with decision support tools or with links from another system or from a PC.

Tip: When your system participates in a network, *USE authority may provide more authority than you expect. For example, with FTP, you can make a copy of a file to another system (including a PC) if you have *USE authority to the file.

Use library security to complement menu security:

This article describes how to set library authority for users of a particular menu.

To access an object in a library, you must have authority both to the object and to the library. Most operations require either *EXECUTE authority or *USE authority to the library.

Depending on your situation, you may be able to use library authority as a simple means for securing objects. For example, assume that for the Order Entry menu example, everyone who has authority to the Order Entry menu can use all of the programs in the ORDERPGM library. Rather than securing individual programs, you can set the public authority to the ORDERPGM library to *EXCLUDE. You can then grant *USE authority to the library to specific user profiles, which will allow them to use the programs in the library. This assumes that public authority to the programs is *USE or greater.

Library authority can be a simple, efficient method for administering object authority. However, you must ensure that you are familiar with the contents of the libraries that you are securing so that you do not provide unintended access to objects.

Secure the integrated file system

The integrated file system provides you with multiple ways to store and view information on the system.

The integrated file system is a part of the i5/OS operating system that supports stream input and output operations. It provides storage management methods that are similar to, and compatible with, personal computer operating systems and UNIX operating systems.

With the integrated file system, all objects on the system can be viewed from the perspective of a hierarchical directory structure. However, in most cases, users view objects in the way that is most common for a particular file system. For example, standard system objects are in the QSYS.LIB file system. Typically, users view these objects from the perspective of libraries. Users typically view objects

in the QDLS file system from the perspective of documents within folders. The “root” (/), QOpenSys, and user-defined file systems present a structure of hierarchical directories.

As a security administrator, you need to understand:

- Which file systems are used on your system
- The unique security characteristics of each file system

The “root” (/) file system acts as a foundation for all other file systems on IBM Systems. At a high level, it provides an integrated view of all of the objects on the system. Other file systems that can exist on IBM Systems provide varying approaches to object management and integration, depending on the underlying purpose of each file system. The QOPT (optical) file system, for example, allows system applications and servers, including the iSeries Access for Windows file server, to access the CD-ROM drive on the system. Similarly, the QFileSvr.400 file system allows applications to access integrated file system data on remote systems. The QLANSrv file server allows access to files stored on Integrated xSeries Server for iSeries or other connected servers in the network.

The security approach for each file system depends on the data that the file system makes available. The QOPT file system, for example, does not provide object-level security because no technology exists to write authority information to a CD-ROM. For the QFileSvr.400 file system, access control occurs at the remote system, where the files are physically stored and managed. For file systems like QLANSrv, the Integrated xSeries Server for iSeries provides access control. Despite the differing security models, many file systems support consistent management of access control through the integrated file system commands, such as Change Authority (CHGAUT) and Change Owner (CHGOWN).

Secure your printer output queue

This article describes the printer output queue setup tasks, explains why they are important, and provides step-by-step instructions for these tasks:

1. Type CRTOUTQ (Create Output Queue) and press F4 (Prompt).
2. Fill in the name of the output queue and the library.
3. Press F10 (Additional parameters).
4. Page down to the find the security information for the output queue.
5. Fill in the information from your Output Queue and Workstation Security form to control who can use and manage the output queue.
6. Press the Enter key and check for confirmation messages.

Possible error	Recovery
You pressed the Enter key instead of F10 on step 3.	Use the Change Output Queue (CHGOUTQ) command to enter additional information.
You created the output queue in the wrong library.	Use the Move Object (MOVOBJ) command to move it to the correct library.

You can now assign printer output to an output queue.

Secure your workstations

After you secure printer output, you should secure your workstations. You authorize workstations just like you authorize other objects on the system. Use the EDTOBJAUT command to give users authority to workstations.

Your system users have PCs on their desks as their workstations. They use tools that run on the PC, and they use the PC to connect to the server. Most methods of connecting a PC to IBM Systems provide more function than workstation emulation. The PC may look like a display to the system and provide the user with interactive signon sessions. In addition, the PC may look to IBM Systems like other computers and provide functions such as file transfer and remote procedure call.

As an IBM Systems security administrator, you need to be aware of the following:

- Functions that are available to PC users who are connected to your system
- IBM Systems resources that PC users can access.

You may want to prevent advanced PC functions, such as file transfer and remote procedure call, if your security scheme is not yet prepared for those functions. Probably, your long-range goal is to allow advanced PC functions while you still protect the information on your system. The topics that follow discuss some of the security issues that are associated with PC access.

Secure workstation data access

Some PC client software uses shared folders to store information on the server. To access system database files, the PC user has a limited, well-defined set of interfaces. With the file transfer capability that is part of most client/server software, the PC user can copy files between the server and the PC. With database access capability; such as a DDM file, remote SQL, or an ODBC driver; the PC user can access data on the server.

In this environment, you can create programs to intercept and evaluate PC-user requests to access server resources. When the requests use a DDM file, you specify the exit program in the distributed data management access (DDMACC) network attribute. For some methods of PC file transfer, you specify the exit program in the client request access (PCSACC) network attribute. Or, you can specify PCSACC (*REGFAC) to use the registration function. When the requests use other server functions to access data, you can use the WRKREGINF command to register exit programs for those server functions.

Exit programs, however, can be difficult to design, and they are rarely foolproof. Exit programs are not a replacement for object authority, which is designed to protect your objects from unauthorized access from any source.

Some client software, such as IBM iSeries Access for Windows, uses the integrated file system to store and access data on IBM Systems. With the integrated file system, the entire server becomes more easily available to PC users. Object authority becomes even more essential. Through the integrated file system, a user with sufficient authority can view a server library as if it is a PC directory. Simple move and copy commands can instantly move data from a system library to a PC directory or vice versa. The system automatically makes the appropriate changes to the format of the data.

Note: You can use an authorization list to control the use of objects in the QSYS.LIB file system.

The strength of the integrated file system is its simplicity for users and developers. With a single interface, the user can work with objects in multiple environments. The PC user does not need special software or APIs to access objects. Instead, the PC user can use familiar PC commands or “point and click” to work with objects directly.

For all systems that have PCs attached, but particularly for systems that have client software that uses the integrated file system, a good object authority scheme is critical. Because security is integrated into the i5/OS product, any request to access data must go through the authority checking process. Authority checking applies to requests from any source and to data access that uses any method.

Object authority with workstation access

When you set up authority for objects, you need to evaluate what that authority provides for the PC user. For example, when a user has *USE authority to a file, the user can view or print data in the file. The user cannot change information in the file or delete the file. For the PC user, viewing is equivalent to reading, which provides sufficient authority for the user to make a copy of a file on the PC. This may not be what you intend.

For some critical files, you may need to set the public authority to *EXCLUDE to prevent downloading. You can then provide another method to view the file on the server, such as using a menu and programs that adopt authority. Another option to prevent downloading is to use an exit program that runs whenever a PC user starts a server function, other than interactive signon.

You can specify an exit program in the PCSACC network attribute by using the Change Network Attribute (CHGNETA) command. Or, you can register exit programs by using the Work with Registration Information (WRKREGINF) command. The method that you use depends on how PCs are accessing data on your system and which client program the PCs use. The exit program (QIBM_QPWFS_FILE_SERV) applies to iSeries Access and Net Server access to integrated file system. It does not prevent access from a PC with other mechanisms, such as FTP or ODBC.

PC software typically provides upload capability also, so that a user can copy data from the PC to a server database file. If you have not set up your authority scheme correctly, a PC user might overlay all of the data in a file with data from a PC. You need to assign *CHANGE authority carefully. Review Appendix D in the iSeries Security Reference to understand what authority is required for file operations.

Users must have *CHANGE authority to sign on at a workstation. If the QLMTSECOFR system value is no (0), the security officer or anyone with *ALLOBJ authority can sign on at any workstation. If the QLMTSECOFR system value is yes (1), use these guidelines to set authority to workstations:

Users allowed to sign on at workstation	Public authority	QSECOFR authority	Individual user authority
All users	*CHANGE	*CHANGE	Not required
Only selected users	*EXCLUDE	No authority	*CHANGE
Selected users and users with authority to all objects	*EXCLUDE	*CHANGE	*CHANGE
All users except users with authority to all objects	*CHANGE	No authority	Not required

As an IBM Systems security administrator, you need to be aware of the following:

- Functions that are available to PC users who are connected to your system
- Resources of IBM Systems that PC users can access.

You may want to prevent advanced PC functions, such as file transfer and remote procedure call, if your security scheme is not yet prepared for those functions. Your long-range goals probably include allowing advanced PC functions while you still protect the information on your system.

Before you restrict access to the system operator message queue, use the EDTOBJAUT command to secure workstations, based on the information in your Output Queue and Workstation Security form.

Object authority with workstation access:

When you set up authority for objects, you need to evaluate what that authority provides for the PC user.

For example, when a user has *USE authority to a file, the user can view or print data in the file. The user cannot change information in the file or delete the file.

For the PC user, viewing provides sufficient authority for the user to make a copy of a file on the PC. This may not be what you intend. For some critical files, you may need to set the public authority to *EXCLUDE to prevent downloading. You can then provide another method to view the file on the server, such as using a menu and programs that adopt authority.

Another option to prevent downloading is to use an exit program that runs whenever a PC user starts a server function, other than interactive signon. You can specify an exit program in the PCSACC network attribute by using the Change Network Attribute (CHGNETA) command. Or, you can register exit programs by using the Work with Registration Information (WRKREGINF) command. The method that you use depends on how PCs are accessing data on your system and which client program the PCs use. The exit program (QIBM_QPWFS_FILE_SERV) applies to iSeries Access and Net Server access to IFS. It does not prevent access from a PC with other mechanisms, such as FTP or ODBC.

Application administration:

Application Administration is an optional component of iSeries Navigator, the graphical user interface (GUI) for the iSeries server.

Application Administration allows system administrators to control the functions or applications available to users and groups on a specific server. This includes controlling the functions available to users that access their server through clients. It is important to note here, that if you access the server from a Windows client, the iSeries server user and not the Windows user determines which functions are available for administration.

For complete documentation on iSeries Navigator Application Administration, refer to Application Administration.

Policy administration

Policies are a tool for administrators to use as they configure software on their client PCs. Policies can restrict which functions and applications a user has access to on the PC. Policies can also suggest or mandate configurations to be used by certain users or certain PCs.

Note: Polices do not offer control over server resources. Policies are not a substitute for server security. Policies can be used to affect how iSeries Access is able to access the server from a particular PC, by a particular user. However, they do not change how server resources can be accessed via other mechanisms.

Policies are stored on a file server. Each time the user signs on to their Windows workstation, the policies that apply to that Windows user are downloaded from the file server. The policies are applied to the registry before the user does anything on the workstation.

Microsoft policies versus application administration:

iSeries Access Express supports two different strategies for implementing administrative control within your network: Microsoft system policies and iSeries Navigator Application Administration. Consider the following when deciding which strategy is best suited for your needs.

Microsoft system policies:

Policies are PC driven, not dependent upon specific OS/400 releases. Policies can apply to PCs as well as Windows users. This means that users refer to the Windows user profile, not the server user profile. Policies can be used to configure as well as to restrict. Policies typically will offer more granularity than Application Administration and can offer a larger breadth of function. This is because a connection to the server is not needed to determine whether the user can use the function or not. Implementing policies is more complicated than implementing Application Administration because the use of the Microsoft system policy editor is required and PCs must be individually configured to download policies.

iSeries Navigator application administration:

Application Administration associates data with the user profile, instead of the Windows profile that Microsoft system policies associate with. Application Administration uses the graphical user interface of iSeries Navigator to administer, which is much easier to use than policy editor. Application Administration info applies to the user regardless of which PC he signs on from. Particular functions within iSeries Navigator can be restricted. Application Administration is preferable if all of the functions you want to restrict are Application Administration-enabled, and if the version of OS/400 being used supports Application Administration.

Prevent ODBC access:

Open database connectivity (ODBC) is a tool that PC applications can use to access iSeries data as if the data is PC data.

The ODBC programmer can make the physical location of the data transparent to the user of the PC application. For more information regarding ODBC security considerations, go to iSeries Access for Windows ODBC security.

Security considerations for workstation session passwords:

This topic discusses the security concerns over passwords being exchanged between workstations and servers.

Typically, when a PC user starts the connection software, such as iSeries Access, the user types the user ID and password for the server once. The password is encrypted and stored in PC memory. Whenever the user establishes a new session to the same server, the PC sends the user ID and password automatically.

Some client/server software also provides the option of bypassing the Sign On display for interactive sessions. The software will send the user ID and encrypted password when the user starts an interactive (5250 emulation) session. To support this option, the QRMTSIGN system value on the server must be set to *VERIFY.

When you choose to allow bypassing the Sign On display, you need to consider the security trade-offs.

Security exposure: For 5250 emulation or any other type of interactive session, the Sign On display is the same as any other display. Although the password is not displayed on the screen when it is typed, the password is sent over the link in unencrypted form just like any other data field. For some types of links, this may provide the opportunity for a would-be intruder to monitor the link and to detect a user ID and password. Monitoring a link by using electronic equipment is often referred to as sniffing. Beginning with V4R4, you can use secure sockets layer (SSL) to encrypt communication between iSeries Access and the iSeries server. This protects your data, including passwords, from sniffing.

When you choose the option to bypass the Sign On display, the PC encrypts the password before it is sent. Encryption avoids the possibility of having a password stolen by sniffing. However, you must ensure that your PC users practice operational security. An unattended PC with an active session to the iSeries system provides the opportunity for someone to start another session without knowing a user ID and password. PCs should be set up to lock when the system is inactive for an extended period, and they should require a password to resume the session.

Even if you do not choose to bypass the Sign On display, an unattended PC with an active session represents a security exposure. By using PC software, someone can start a server session and access data, again without knowing a user ID and a password. The exposure with 5250 emulation is somewhat greater because it requires less knowledge to start a session and begin accessing data.

You also need to educate your users about the effect of disconnecting their iSeries Access session. Many users assume, logically but incorrectly, that the disconnect option completely stops their connection to the

server. In fact, when a user selects the option to disconnect, the server makes the user's session available for another user. However, the client's connection to the server is still open. Another user could walk up to the unprotected PC and get access to server resources without ever entering a user ID and password.

You can suggest two options for your users who need to disconnect their sessions:

- Ensure that their PCs have a lockup function that requires a password. This makes an unattended PC unavailable to anyone who does not know the password.
- To completely disconnect a session, either log off Windows or restart (reboot) the PC. This ends the session to the iSeries.

You also need to educate your users about a potential security exposure when they use iSeries Access for Windows. When a user specifies a UNC (universal naming convention) to identify an iSeries resource, the Win95 or NT client builds a network connection to link to the server. Because the user specifies a UNC, the user does not see this as a mapped Network Drive. Often, the user is not even aware of the existence of the network connection. However, this network connection represents a security exposure on an unattended PC because the server appears in the directory tree on the PC. If the user's session has a powerful user profile, server resources might be exposed on an unattended PC. As with the previous example, the remedy is to ensure both that users understand the exposure and that they use their PC's lockup function.

Protect the server from remote commands and procedures:

This topic explains why you need to consider how remote commands and procedures can be run on your server.

A knowledgeable PC user with software such as iSeries Access can run commands on the server without going through the Sign On display. The following are several methods that are available for PC users to run server commands. Your client/server software determines the methods that your PC users have available to them.

- A user can open a DDM file and use the remote command function to run a command.
- Some software, such as iSeries Access optimized clients, provides the remote command function through Distributed Program Call (DPC) APIs, without the use of DDM.
- Some software, such as remote SQL and ODBC, provides a remote command function without either DDM or DPC.

For client/server software that uses DDM for remote command support, you can use the DDMACC network attribute to prevent remote commands completely. For client/server software that uses other server support, you can register exit programs for the server. If you want to allow remote commands, you must make sure that your object authority scheme protects your data adequately. Remote command capability is equivalent to giving a user a command line. In addition, when iSeries receives a remote command through DDM, the system does not enforce the user profiles Limited capability (LMTCPB) setting.

Protect workstations from remote commands and procedures:

IBM iSeries Access for Windows provides the capability of receiving remote commands on the PC.

You can use the Run Remote Command (RUNRMTCMD) command on the server to run a procedure on an attached PC. The RUNRMTCMD capability is a valuable tool for system administrators and help-desk personnel. However, it also provides the opportunity for damaging PC data, either deliberately or accidentally.

PCs do not have the same object authority functions as iSeries servers. Your best protection against problems with the RUNRMTCMD command is to carefully restrict the system users who have access to the command. IBM iSeries Access for Windows provides the capability to register which users can run

remote commands on a specific PC. When the connection is via TCP/IP, you can use the properties control panel on the client to control remote-command access. You can authorize users by user ID or by the remote system name. When the connection is via SNA, some client software provides the capability to set up security for the conversation. With other client software, you simply choose whether or not to set up the incoming-command capability.

For each combination of client software and connection type (such as TCP/IP or SNA), you need to review the potential for incoming-commands to attached PCs. Consult the client documentation by searching for "incoming command" or "RUNRMTCMD". Be prepared to advise your PC users and network administrators about the correct (secure) way to configure clients to permit or prevent this capability.

Gateway servers:

Your system may participate in a network with an intermediate or gateway server between the iSeries system and the PCs.

For example, your iSeries system might be attached to a LAN with a PC server that has PCs that are attached to the server. The security issues in this situation depend on the capabilities of the software that is running on the gateway server.With some software, your iSeries system will not know about any users (such as USERA or USERC) who are downstream from the gateway server. The server will sign on to the system as a single user (USERGTW). It will use the USERGTW user ID to handle all requests from downstream users. A request from USERA will look to the server like a request from user USERGTW.

If this is the case, you must rely on the gateway server for security enforcement. You must understand and manage the security capabilities of the gateway server. From an iSeries server perspective, every user has the same authority as the user ID that the gateway server uses to start the session. You might think of this as equivalent to running a program that adopts authority and provides a command line.

With other software, the gateway server passes requests from individual users to iSeries servers. The iSeries server knows that USERA is requesting access to a particular object. The gateway is almost transparent to the system.

If your system is in a network that has gateway servers, you need to evaluate how much authority to provide to the user IDs that are used by the gateway servers. You also need to understand the following:

- The security mechanisms that the gateway servers enforce.
- How downstream users will appear to your iSeries system.

Wireless LAN communications:

Some clients might use the iSeries Wireless LAN to communicate to your system without wires.

The system's wireless LAN uses radio-frequency communications technology. As a security administrator, you should be aware of the following security characteristics of system wireless LAN products:

- These wireless LAN products use spread spectrum technology. This same technology has been used by the government in the past to secure radio transmissions. To someone who attempts to electronically monitor for data transmissions, the transmissions appear to be noise rather than an actual transmission.
- The wireless connection has three security-relevant configuration parameters:
  - Data rate (two possible data rates)
  - Frequency (five possible frequencies)
  - System identifier (8 million possible identifiers)

  These configuration elements combine to provide 80 million possible configurations, which makes a hacker's likelihood of guessing the correct configuration extremely slim.

- Just like with other communications methods, the security of wireless communications is affected by the security of the client device. The system ID information and other configuration parameters are in a file on the client device and should be protected.
- If a wireless device is lost or stolen, normal server security measures, such as signon passwords and object security, provide protection when an unauthorized user attempts to use the lost or stolen unit to access your system.
- If a wireless client unit is lost or stolen, you should consider changing the system ID information for all users, access points, and systems. Think of this as changing the locks on your doors if a set of keys is stolen.
- You might want to partition your server into groups of clients that have unique system IDs. This limits the impact if a unit is lost or stolen. This method works only if you can confine a group of users to a specific portion of your installation.
- Unlike wired LAN technology, wireless LAN technology is proprietary. Therefore, no electronic sniffers are publicly available for these wireless LAN products. A sniffer is an electronic device that performs unauthorized monitoring of a transmission.

Set up network security

The following topics provide security recommendations for TCP/IP protocols, such as FTP, BOOTP, and VPN; and for APPC.

Set up APPC security

This group of articles discuss various aspects of setting up security for APPC sessions.

There are several aspects of security for an i5/OS system, communicating with each other using APPC and APPN:

- **Physical security** surrounding the systems, communication lines, and display stations that can be configured.
- **Location security** that verifies the identity of other systems in the network.
- **User security** that verifies the identity and rights of users to issue commands on their local system and remote systems when you specify *NONE for the location password (LOCPWD) parameter during APPC configuration.
- **Resource security** that controls user access to particular resources, such as confidential databases remote system when a session is being established.
- **Session-level security** which is achieved by specifying a password on the LOCPWD parameter during configuration. The i5/OS system uses the password to validate the identity of the remote system when a session is being established.

When the system is using level 10 security, APPC connects to the network as a nonsecure system. The i5/OS system does not validate the identity of a remote system when a session is being established and does not require transaction security on incoming program start requests.

If the i5/OS system is the remote system and is using level 20 or above, APPC connects to the network as a secure system.

Restrict APPC sessions:

Use object authority to control access to APPC sessions.

As security administrator on a source system, you can use object authority to control who can attempt to access other systems. Set the public authority for APPC device descriptions to *EXCLUDE and give *CHANGE authority to specific users. Use the QLMTSECOFR system value to prevent users with *ALLOBJ special authority from using APPC communications.

As security administrator on a target system, you can also use authority to APPC devices to prevent users from starting an APPC session on your system. However, you need to understand what user ID will be attempting to access the APPC device description.

Tip: You can use the Print Publicly Authorized Objects (PRTPUBAUT *DEVD) command and the Print Private Authorities (PRTPVTAUT *DEVD) command to find out who has authority to device descriptions on your system.

When your system uses APPN, it automatically creates a new APPC device when no existing device is available for the route that the system has chosen. One method for restricting access to APPC devices on a system that is using APPN is to create an authorization list. The authorization list contains the list of users who should be authorized to APPC devices. You then use the Change Command Default (CHGCMDDFT) command to change the CRTDEVAPPC command. For the authority (AUT) parameter on the CRTDEVAPPC command, set the default value to the authorization list that you created.

You use the location password (**LOCPWD**) parameter in the APPC device description to validate the identity of another system that is requesting a session on your system, on behalf of a user or an application. The location password can help you detect an imposter system.

When you use location passwords, you must coordinate with security administrators for other systems in the network. You must also control who can create or change APPC device descriptions and configuration lists. The system requires *IOSYSCFG special authority to use the commands that work with APPC devices and configuration lists.

Tip: When you use APPN, the location passwords are stored in the QAPPNRMT configuration list rather than in device descriptions.

Target system assignment of user profiles for jobs:

When a user requests an APPC job on another system, the request has a mode name associated with it. The mode name might come from the user's request, or it might be a default value from the network attributes of the source system.

The target system uses the mode name and the APPC device name to determine how the job will run. The target system searches the active subsystems for a communications entry that is the best match for the APPC device name and the mode name.

The communications entry specifies what user profile the system will use for SECURITY(NONE) requests. An example of a communications entry in a subsystem description:

```
          Display Communications Entries

Subsystem description: QCMN Status: ACTIVE

                 Job                     Default Max
Device   Mode    Description   Library   User    Active
*ALL     *ANY    *USRPRF                 *SYS    *NOMAX
*ALL     QPCSUPP *USRPRF                 *NONE   *NOMAX
```

The following table shows the possible values for the default user parameter in a communications entry:

Table 109. Possible values for the default user parameter

Value	Result
***NONE**	No default user is available. If the source system does not supply a user ID on the request, the job will not run.
***SYS**	Only IBM-supplied programs (system jobs) will run. No user applications will run.

Table 109. Possible values for the default user parameter (continued)

Value	Result
user-name	If the source system does not send a user ID, the job runs under this user profile.

You can use the Print Subsystem Description (PRTSBSDAUT) command to print a list of all subsystems that have communications entries with a default user profile.

Display station passthrough options:

Display station passthrough is an example of an application that uses APPC communications. You can use display station passthrough to sign on to another system that is connected to your system through a network.

The following table shows examples of passthrough requests (STRPASTHR command) and how the target system handles them. For display station passthrough, the system uses the basic elements of APPC communications and the remote sign-on (QRMTSIGN) system value.

Table 110. Sample pass-through sign-on requests

Values on STRPASTHR command		Target system		
User ID	**Password**	**SECURELOC value**	**QRMTSIGN value**	**Result**
*NONE	*NONE	Any	Any	The user must sign on the target system.
A user profile name	Not entered	Any	Any	The request fails.
*CURRENT	Not entered	*NO	Any	The request fails.
		*YES	*SAMEPRF	An interactive job starts with the same user profile name as the user profile on the source system. No password is passed to the remote system. The user profile name must exist on the target system.
			*VERIFY	
			*FRCSIGNON	The user must sign on the target system.
		*VFYENCPWD	*SAMEPRF	An interactive job starts with the same user profile name as the user profile on the source system. The source system retrieves the user's password and sends it to the remote system. The user profile name must exist on the target system.
			*VERIFY	
			*FRCSIGNON	The user must sign on the target system.

Table 110. Sample pass-through sign-on requests (continued)

Values on STRPASTHR command		Target system		
User ID	**Password**	**SECURELOC value**	**QRMTSIGN value**	**Result**
*CURRENT (or the name of the current user profile for the job)	Entered	Any	*SAMEPRF	An interactive job starts with the same user profile name as the user profile on the source system. The password is sent to the remote system. The user profile name must exist on the target system.
			*VERIFY	
			*FRCSIGNON	The user must sign on the target system.
A user profile name (a name different from the current user profile for the job)	Entered	Any	*SAMEPRF	The request fails.
			*VERIFY	An interactive job starts with the same user profile name as the user profile on the source system. The password is sent to the remote system. The user profile name must exist on the target system.
			*FRCSIGNON	An interactive job starts with the specified user profile name. The password is sent to the remote system. The user profile name must exist on the target system.

Avoid unexpected device assignments:

When a failure occurs on an active device, the system attempts to recover. In some circumstances, when the connection is broken, another user can unintentionally reestablish the session that had the failure.

For example, assume that USERA powered off a workstation without signing off. USERB could power on the workstation and restart USERA's session without signing on. To prevent this possibility, set the Device I/O Error Action (QDEVRCYACN) system value to *DSCMSG. When a device fails, the system will end the user's job.

Control remote commands and batch jobs:

Several options are available to help you control which remote commands and jobs can run on your system.

You can use the network job action (JOBACN) network attribute to prevent network jobs from being submitted or to prevent them from running automatically.

If your system uses distributed data management (DDM), you can do the following:

- Restrict access to DDM files to prevent users from using the Submit Remote Command (SBMRMTCMD) command from another system. To use the SBMRMTCMD, the user must be able to open a DDM file. You also need to restrict the ability to create DDM files.
- Specify an exit program for the DDM request access (DDMACC) system value. In the exit program, you can evaluate all DDM requests before allowing them.

You can specify explicitly which program requests can run in a communications environment by removing the PGMEVOKE routing entry from subsystem descriptions. The PGMEVOKE routing entry allows the requester to specify the program that runs. When you remove this routing entry from subsystem descriptions, such as the QCMN subsystem description, you must add routing entries for the communications requests that need to run successfully.

For each request that you want to allow, you can add a routing entry with the compare value and the program name both equal to the program name. When you use this method, you need to understand the work management environment on your system and the types of communications requests that occur on your system. If possible, you should test all types of communications requests to ensure that they work properly after you change the routing entries. When a communications request does not find an available routing entry, you receive a CPF1269 message. Another alternative is to set the public authority to *EXCLUDE for the transaction programs that you do not want to run on your system.

Evaluate your APPC configuration:

You can use the Print Communications Security (PRTCMNSEC) command or menu options to print the security-relevant values in your APPC configuration.

The topics that follow describe the information on the reports.

Relevant parameters for APPC devices:

This articles shows examples of reports for device descriptions and configuration lists.

This table shows an example of the Communications Information Report for device descriptions.

```
Communications Information (Full Report)
                                                SYSTEM4
Object type . . . . . . . . . . : *DEVD
                                                          Pre       SNUF
Object  Object   Device   Secure  Location  APPN   Single   Establish Program
Name    Type    Category Location Password Capable Session  Session   Start

CDMDEV1 *DEVD   *APPC      *NO       *NO     *NO     *YES     *NO
CDMDEV2 *DEVD   *APPC      *NO       *NO     *NO     *YES     *NO
```

Figure 3. APPC device descriptions—sample report

This table shows an example of the report for configuration lists.

```
             Display Configuration List              Page 1

Configuration list . . . . . . . . . : QAPPNRMT
Configuration list type  . . . . . . : *APPNRMT
Text . . . . . . . . . . . . . . . . :
-----------------APPN Remote Locations------------------
          Remote             Remote    Control
Remote    Network  Local     Control   Point    Secure
Location  ID       Location  Point     Net ID     Loc
SYSTEM36  APPN     SYSTEM4   SYSTEM36  APPN       *NO
SYSTEM32  APPN     SYSTEM4   SYSTEM32  APPN       *NO
SYSTEMU   APPN     SYSTEM4   SYSTEM33  APPN       *YES
SYSTEMJ   APPN     SYSTEM4   SYSTEMJ   APPN       *NO
SYSTEMR2  APPN     SYSTEM4   SYSTEM1   APPN       *NO
-------------------------APPN Remote Locations--------------------------
          Remote                                     Local   Pre-
Remote    Network  Local     Single    Number of    Control established
Location  ID       Location  Session Conversations Point   Session
SYSTEM36  APPN     SYSTEM4     *NO         10        *NO      *NO
SYSTEM32  APPN     SYSTEM4     *NO         10        *NO      *NO
```

Figure 4. Configuration list report example

Parameters for APPC controllers:

This article shows an example of the Communications Information Report for controller descriptions.

```
                Communications Information (full report)

Object type . . . . . . . . . . : *CTLD

Object Object Controller  Auto   Switched    Call     APPN      CP    Disconnect Delete  Device
Name   Type   Category   Create Controller Direction Capable Sessions   Timer    Seconds Name
CTL01  *CTLD  *APPC       *YES     *YES       *DIAL    *YES     *YES       0      1440   AARON
CTL02  *CTLD  *APPC       *YES     *YES       *DIAL    *YES     *YES       0      1440   BASIC
```

Figure 5. APPC controller descriptions—sample report

Parameters for line descriptions:

This article shows an example of the Communications Information Report for line descriptions.

```
              Communications Information (Full Report)

Object type . . . . . . . . . . : *LIND
Auto
Object  Object  Line      Auto    Delete   Auto    Auto
Name    Type    Category  Create  Seconds  Answer  Dial
LINE01  *LIND   *SDLC      *NO       0       *NO     *NO
LINE02  *LIND   *SDLC      *NO       0       *YES    *NO
LINE03  *LIND   *SDLC      *NO       0       *NO     *NO
LINE04  *LIND   *SDLC      *NO       0       *YES    *NO
```

Figure 6. APPC line descriptions—sample report

Set up TCP/IP security

The following information guides you through the process of setting up TCP/IP security.

Security considerations for using SLIP

TCP/IP support includes Serial Interface Line Protocol (SLIP).

SLIP provides low-cost point-to-point connectivity. A SLIP user can connect to a LAN or a WAN by establishing a point-to-point connection with a system that is part of the LAN or WAN. SLIP runs on an asynchronous connection. You can use SLIP for dial-up connection to and from iSeries servers.

For example, you might use SLIP to dial in from your PC to an iSeries system. After the connection is established, you can use the TELNET application on your PC to connect to the iSeries TELNET server. Or, you can use the FTP application to transfer files between the two systems.

No SLIP configuration exists on your system when it ships. Therefore, if you do not want SLIP (and dial-up TCP/IP) to run on your system, do not configure any configuration profiles for SLIP. You use the Work with TCP/IP Point-to-Point (WRKTCPPTP) command to create SLIP configurations. You must have *IOSYSCFG special authority to use the WRKTCPPTP command.

If you want SLIP to run on your system, you create one or more SLIP (point-to-point) configuration profiles. You can create configuration profiles with the following operating modes:

- Dial in (*ANS)
- Dial out (*DIAL)

Note: A user profile is system object that allows signon. Every system job must have a user profile to run. A configuration profile stores information that is used to establish a SLIP connection with an iSeries system. When you start a SLIP connection to iSeries servers, you are simply establishing a link. You have not yet signed on and started an iSeries server job. Therefore, you do not necessarily need a user profile to start a SLIP connection to iSeries servers. However, as you will see in the discussions that follow, the SLIP configuration profile may require a user profile to determine whether to allow the connection.

Secure dial-in SLIP connections:

Before someone can establish a dial-in connection to your system with SLIP, you must start a SLIP *ANS configuration profile.

To create or change a SLIP configuration profile, you use the Work with TCP/IP Point-to-Point (WRKTCPPTP) command. To start a configuration profile, you use either the Start TCP/IP Point-to-Point (STRTCPPTP) command or an option from the WRKTCPPTP display. When your system ships, the public authority for the STRTCPPTP and ENDTCPPTP commands are *EXCLUDE. The options to add, change, and delete SLIP configuration profiles are available only if you have *IOSYSCFG special authority. As security administrator, you can use both command authority and special authority determine who can set up your system to allow dial-in connections.

If you want to validate systems that dial in to your system, then you want the requesting system to send a user ID and a password. Your system can then verify the user ID and password. If the user ID and password are not valid, your system can reject the session request. To set up dial-in validation, do the following:

1. Create a user profile that the requesting system can use to establish the connection. The user ID and password that the requester sends must match this user profile name and password. **Note:** For the system to perform password validation, the QSECURITY system value must be set to 20 or higher. As additional protection, you probably want to create user profiles specifically for establishing SLIP connections. The user profiles should have limited authority on the system. If you do not plan to use the profiles for any function except establishing SLIP connections, you can set the following values in the user profiles:An initial menu (INLMNU) of *SIGNOFF, An initial program (INLPGM) of *NONE, and Limit capabilities (LMTCPB) of *YES. These values prevent anyone from signing on interactively with the user profile.
2. Create an authorization list for the system to check when a requester tries to establish a SLIP connection. **Note:** You specify this authorization list in the System access authorization list field when you create or change the SLIP profile.

3. Use the Add Authorization Entry (ADDAUTLE) command to add the user profile that you created in step 1 to the authorization list. You can create a unique authorization list for each point-to-point configuration profile, or you can create an authorization list that several configuration profiles share.
4. Use the WRKTCPPTP command to set up a TCP/IP point-to-point *ANS profile that has the following characteristics:
   a. The configuration profile must use a connection dialog script that includes the user-validation function. User validation includes accepting a user ID and password from the requester and validating them. The system ships with several sample dialog scripts that provide this function.
   b. The configuration profile must specify the name of the authorization list that you created in step 2. The user ID that the connection dialog script receives must be in the authorization list.

Keep in mind that the value of setting up dial-in security is affected by the security practices and capabilities of the systems that dial in. If you require a user ID and password, then the connection dialog script on the requesting system must send that user ID and password. Some systems, such as iSeries servers, provide a secure method for storing the user IDs and passwords. Other systems store the user ID and password in the script which might be accessible to anyone who knows where to find the script on the system.

Because of the differing security practices and capabilities of your communications partners, you might want to create different configuration profiles for different requesting environments. You use STRTCPPTP command to set your system up to accept a session for a specific configuration profile. You can start sessions for some configuration profiles only at certain times of the day, for example. You might use security auditing to log the activity for the associated user profiles.

Prevent dial-in users from accessing other systems:

Depending on your system and network configuration, a user who starts a SLIP connection might be able to access another system in your network without signing on to your system.

For example, a user could establish a SLIP connection to your system. Then the user could establish an FTP connection to another system in your network that does not allow dial-in.

You can prevent a SLIP user from accessing other systems in your network by specifying N (No) for the Allow IP datagram forwarding field in the configuration profile. This prevents a user from accessing your network before the user logs on to your system. However, after the user has successfully logged on to your system, the datagram forwarding value has no effect. It does not limit the user's ability to use a TCP/IP application on your iSeries system (such as FTP or TELNET), to establish a connection with another system in your network.

Control dial-out sessions:

Before someone can use SLIP to establish a dial-out connection from your system, you must start a SLIP *DIAL configuration profile.

To create or change a SLIP configuration profile, you use the WRKTCPPTP command. To start a configuration profile, you use either the Start TCP/IP Point-to-Point (STRTCPPTP) command or an option from the WRKTCPPTP display. When your system ships, the public authority for the STRTCPPTP and ENDTCPPTP commands are *EXCLUDE. The options to add, change, and delete SLIP configuration profiles are available only if you have *IOSYSCFG special authority. As security administrator, you can use both command authority and special authority determine who can set up your system to allow dial-out connections.

Secure dial-out sessions:

Users on your iSeries system might want to establish dial-out connections to systems that require user validation.

The connection dialog script on your iSeries server must send a user ID and a password to the remote system. iSeries servers provide a secure method for storing that password. The password does not need to be stored in the connection dialog script.

Note:

1. your system decrypts the password before sending it. SLIP passwords, like FTP and TELNET passwords, are sent unencrypted ("in the clear"). However, unlike with FTP and TELNET, the SLIP password is sent before the systems establish TCP/IP mode.
2. Because SLIP uses a point-to-point connection in asynchronous mode, the security exposure when sending unencrypted passwords is different from the exposure with FTP and TELNET passwords. Unencrypted FTP and TELNET passwords might be sent as IP traffic on a network and are, therefore, vulnerable to electronic sniffing. The transmission of your SLIP password is as secure as the telephone connection between the two systems. 2. The default file for storing SLIP connection dialog scripts is QUSRSYS/QATOCPPSCR. The public authority for this file is *USE, which prevents public users from changing the default connection dialog scripts.

When you create a connection profile for a remote session that requires validation, do the following:

1. Ensure that the Retain Server Security Data (QRETSVRSEC) system value is 1 (Yes). This system value determines whether you will allow passwords that can be decrypted to be stored in a protected area on your system.
2. Use the WRKTCPPTP command to create a configuration profile that has the following characteristics:
   a. For the mode of the configuration profile, specify *DIAL.
   b. For the Remote service access name, specify the user ID that the remote system expects. For example, if you are connecting to another iSeries server, specify the user profile name on that iSeries server.
   c. For the Remote service access password, specify the password that the remote system expects for this user ID. On your iSeries server, this password is stored in a protected area in a form that can be decrypted. The names and passwords that you assign for configuration profiles are associated with the QTCP user profile. The names and passwords are not accessible with any user commands or interfaces. Only registered system programs can access this password information.

      Note: Keep in mind that the passwords for your connection profiles are not saved when your save the TCP/IP configuration files. To save SLIP passwords, you need to use the Save Security Data (SAVSECDTA) command to save the QTCP user profile.
   d. For the connection dialog script, specify a script that sends the user ID and password. The system ships with several sample dialog scripts that provide this function. When the system runs the script, the system retrieves the password, decrypts it, and sends it to the remote system.

Security considerations for using point-to-point protocol

Point-to-point protocol (PPP) is available as part of TCP/IP.

PPP is an industry standard for point-to-point connections that provides additional function over what is available with SLIP. With PPP, your iSeries server can have high-speed connections directly to an Internet Service Provider or to other systems in an intranet or extranet. Remote LANs can realistically make dial-in connections to your iSeries server.

Remember that PPP, like SLIP, provides a network connection to your iSeries server. A PPP connection essentially brings the requester to your system's door. The requester still needs a user ID and password to enter your system and connect to a TCP/IP server like TELNET or FTP. Following are security considerations with this new connection capability:

Note: You configure PPP by using iSeries Navigator on an IBM iSeries Access for Windows workstation.

- PPP provides the ability to have dedicated connections (where the same user always has the same IP address). With a dedicated address, you have the potential for IP spoofing (an imposter system that pretends to be a trusted system with a known IP address). However, the enhanced authentication capabilities that PPP provides help protect against IP spoofing.
- With PPP, as with SLIP, you create connection profiles that have a user name and an associated password. However, unlike SLIP, the user does not need to have a valid user profile and password. The user name and password are not associated with a user profile. Instead, validation lists are used for PPP authentication. Additionally, PPP does not require a connection script. The authentication (exchange of user name and password) is part of the PPP architecture and happens at a lower level than with SLIP.
- With PPP, you have the option to use CHAP (challenge handshake authentication protocol). You will no longer need to worry about an eavesdropper sniffing passwords because CHAP encrypts user names and passwords.

  Your PPP connection uses CHAP only if both sides have CHAP support. During the exchange signals to set up communications between two modems, the two systems negotiate. For example, if SYSTEMA supports CHAP and SYSTEMB does not, SYSTEMA can either deny the session or agree to use an unencrypted user name and password. Agreeing to use an unencrypted user name and password is referred to as negotiating down.

  The decision to negotiate down is a configuration option. On your intranet, for example, where you know that all your systems have CHAP capability, you should configure your connection profile so that it will not negotiate down. On a public connection where your system is dialing out, you might be willing to negotiate down. The connection profile for PPP provides the ability to specify valid IP addresses. You can, for example, indicate that you expect a specific address or range of addresses for a specific user.

  This capability, together with the ability for encrypted passwords, provides further protection against spoofing. As additional protection against spoofing or piggy-backing on an active session, you can configure PPP to rechallenge at designated intervals. For example, while a PPP session is active, your iSeries server might challenge the other system for a user and password. It does this every 15 minutes to ensure that `it is the same connection profile.

  The end-user will not be aware of this rechallenge activity. The systems exchange names and passwords below the level that the end-user sees. With PPP, it is realistic to expect that remote LANs might establish a dial-in connection to your iSeries server and to your extended network. In this environment, having IP forwarding turned on is probably a requirement. IP forwarding has the potential to allow an intruder to roam through your network. However, PPP has stronger protections (such as encryption of passwords and IP address validation). This makes it less likely that an intruder can establish a network connection in the first place.

Security considerations for using Bootstrap Protocol server

Bootstrap Protocol (BOOTP) provides a dynamic method for associating workstations with servers and assigning workstation IP addresses and initial program load (IPL) sources.

BOOTP is a TCP/IP protocol used to allow a media-less workstation (client) to request a file containing initial code from a server on the network. The BOOTP server listens on the well known BOOTP server port 67. When a client request is received, the server looks up the IP address defined for the client and returns a reply to the client with the client's IP address and the name of the load file. The client then initiates a TFTP request to the server for the load file. The mapping between the client hardware address and IP address is kept in the BOOTP table on the system.

Prevent BOOTP access:

If you do not have any thin clients attached to your network, you do not need to run the BOOTP server on your system.

It can be used for other devices, but the preferred solution for those devices is to use DHCP. Do the following to prevent the BOOTP server from running:

1. To prevent BOOTP server jobs from starting automatically when you start TCP/IP, type the following: `CHGBPA AUTOSTART(*NO)`

   Note:

   a. AUTOSTART(*NO) is the default value.

   b. "Control which TCP/IP servers start automatically" on page 120 provides more information about controlling which TCP/IP servers start automatically.

2. To prevent someone from associating a user application, such as a socket application, with the port that the system normally uses for BOOTP, do the following:

   Note: Because DHCP and BOOTP use the same port number, this will also inhibit the port that is used by DHCP. Do not restrict the port if you want to use DHCP.

   a. Type GO CFGTCP to display the Configure TCP/IP menu.

   b. Select option 4 (Work with TCP/IP port restrictions).

   c. On the Work with TCP/IP Port Restrictions display, specify option 1 (Add).

   d. For the lower port range, specify 67.

   e. For the upper port range, specify *ONLY.

   Note:

   a. The port restriction takes effect the next time that you start TCP/IP. If TCP/IP is active when you set the port restrictions, you should end TCP/IP and start it again.

   b. RFC1700 provides information about common port number assignments.

3. For the protocol, specify *UDP.

4. For the user profile field, specify a user profile name that is protected on your system. A protected user profile is a user profile that does not own programs that adopt authority and does not have a password that is known by other users. By restricting the port to a specific user, you automatically exclude all other users.

Secure the BOOTP server:

The BOOTP server does not provide direct access to your iSeries system, and thus represents a limited security exposure.

Your primary concern as a security administrator is to ensure that the correct information is associated with the correct thin client. In other words, a mischief-maker could alter the BOOTP table and cause your thin clients to work incorrectly or not at all.

To administer the BOOTP server and the BOOTP table, you must have *IOSYSCFG special authority. You need to carefully control the user profiles that have *IOSYSCFG special authority on your system.

Security considerations for using DHCP server

These topics discuss methods for securing the DHCP server for authorized users and preventing access to the DHCP server.

Dynamic host configuration protocol (DHCP) provides a framework for passing configuration information to hosts on a TCP/IP network. For your client workstations, DHCP can provide a function similar to auto configuration. A DHCP-enabled program on the client workstation broadcasts a request for configuration information. If the DHCP server is running on your system, the server responds to the request by sending the information that the client workstation needs to correctly configure TCP/IP.

You can use DHCP to make it simpler for users to connect to your system for the first time. This is because the user does not need to enter TCP/IP configuration information. You can also use DHCP to reduce the number of internal TCP/IP addresses that you need in a subnetwork. The DHCP server can temporarily allocate IP addresses to active users (from its pool of IP addresses).

For thin clients, you can use DHCP in place of BOOTP. DHCP provides more function than BOOTP, and it can support dynamic configuration of both thin clients and PCs.

Prevent DHCP access:

This article discusses the steps for preventing users from accessing the DHCP server.

If you do not want anyone to use the DHCP server on your system, do the following:

1. To prevent DHCP server jobs from starting automatically when you start TCP/IP, type the following:
   `CHGDHCPA AUTOSTART(*NO)`

   Note: AUTOSTART(*NO) is the default value.
2. To prevent someone from associating a user application, such as a socket application, with the port that the system normally uses for DHCP, do the following:
   a. Type `GO CFGTCP` to display the **Configure TCP/IP** menu.
   b. Select option **4** (Work with TCP/IP port restrictions).
   c. On the Work with TCP/IP Port Restrictions display, specify option **1** (Add).
   d. For the lower port range, specify `67`.
   e. For the upper port range, specify `68`.

      Note: The port restriction takes effect the next time that you start TCP/IP. If TCP/IP is active when you set the port restrictions, you should end TCP/IP and start it again.
   f. For the protocol, specify `*UDP`.
   g. For the user profile field, specify a user profile name that is protected on your system. (A protected user profile is a user profile that does not own programs that adopt authority and does not have a password that is known by other users.) By restricting the port to a specific user, you automatically exclude all other users.

Secure the DHCP server:

This article provides recommendations for securing the DHCP server.

Following are security considerations when you choose to run DHCP on your system:

- Restrict the number of users who have authority to administer DHCP. Administering DHCP requires the following authority:
  - *IOSYSCFG special authority
  - *RW authority to the following files:

    ```
    /QIBM/UserData/OS400/DHCP/dhcpsd.cfg
    /QIBM/UserData/OS400/DHCP/dhcprd.cfg
    ```
- Evaluate how physically accessible your LAN is. Could an outsider easily walk into your location with a laptop and physically connect it to your LAN? If this is an exposure, DHCP provides the capability to create a list of clients (hardware addresses) that the DHCP server will configure. When you use this feature, you remove some of the productivity benefit that DHCP provides to your network administrators. However, you prevent the system from configuring unknown workstations.
- If possible, use a pool of IP addresses that is reusable (not architected for the Internet). This helps prevent a workstation from outside your network from gaining usable configuration information from the server.

- Use the DHCP exit points if you need additional security protection. Following is an overview of the exit points and their capabilities.

  Port entry
  The system calls your exit program whenever it reads a data packet from port 67 (the DHCP port). Your exit program receives the full data packet. It can decide whether the system should process or discard the packet. You can use this exit point when existing DHCP screening features are not sufficient for your needs.

  Address assignment
  The system calls your exit program whenever DHCP formally assigns an address to a client.

  Address release
  The system calls your exit program whenever DHCP formally releases an address and places it back in the address pool.

Security considerations for using TFTP server

These articles discuss methods for securing the TFTP server for authorized users and preventing access to the TFTP server.

Trivial file transfer protocol (TFTP) provides basic file transfer with no user authentication. TFTP works with either Bootstrap Protocol (BOOTP) or Dynamic Host Configuration Protocol (DHCP).

The client connects initially to either the BOOTP server or the DHCP server. The BOOTP server or the DHCP server replies with the client's IP address and the name of the load file. The client then initiates a TFTP request to the server for the load file. When the client completes downloading of the load file, it ends the TFTP session.

Prevent TFTP access:

This article discusses the steps for preventing users from accessing the TFTP server.

If you do not have any thin clients attached to your network, you probably do not need to run the TFTP server on your system. Do the following to prevent the TFTP server from running:

1. To prevent TFTP server jobs from starting automatically when you start TCP/IP, type the following: `CHGTFTPA AUTOSTART(*NO)`

   AUTOSTART(*NO) is the default value.
2. To prevent someone from associating a user application, such as a socket application, with the port that the system normally uses for TFTP, do the following:
   a. Type `GO CFGTCP` to display the **Configure TCP/IP** menu.
   b. Select option **4** (Work with TCP/IP port restrictions).
   c. On the Work with TCP/IP Port Restrictions display, specify option **1** (Add).
   d. For the lower port range, specify `69`.
   e. For the upper port range, specify `*ONLY`.

      Note: The port restriction takes effect the next time that you start TCP/IP. If TCP/IP is active when you set the port restrictions, you should end TCP/IP and start it again.
   f. For the protocol, specify `*UDP`.
   g. For the user profile field, specify a user profile name that is protected on your system. (A protected user profile is a user profile that does not own programs that adopt authority and does not have a password that is known by other users.) By restricting the port to a specific user, you automatically exclude all other users.

Secure the TFTP server:

This article provides recommendations for securing the TFTP server.

By default, the TFTP server provides very limited access to your system. It is specifically configured to provide the initial code for thin clients. As a security administrator, you should be aware of the following characteristics of the TFTP server:

- The TFTP server does not require authentication (a user ID and password). All TFTP jobs run under the QTFTP user profile. The QTFTP user profile does not have a password. Therefore, it is not available for interactive sign-on. The QTFTP user profile does not have any special authorities, nor is it explicitly authorized to system resources. It uses public authority to access the resources that it needs for the thin clients.
- When the TFTP server arrives, it is configured to access the directory that contains thin client information. You must have *PUBLIC or QTFTP authorized to read or write to that directory. To write to the directory you must have *CREATE specified on the **Allow file writes** parameter of the CHGTFTPA command. To write to an existing file you must have the *REPLACE specified on the **Allow file writes** parameter of CHGTFTPA. *CREATE allows you to replace existing files or create new files. *REPLACE only allows you to replace existing files.

  A TFTP client cannot access any other directory unless you explicitly define the directory with the Change TFTP Attributes (CHGTFTPA) command. Therefore, if a local or remote user does attempt to start a TFTP session to your system, the user's ability to access information or cause damage is extremely limited.
- If you choose to configure your TFTP server to provide other services in addition to handling thin clients, you can define an exit program to evaluate and authorize every TFTP request. The TFTP server provides a request validation exit similar to the exit that is available for the FTP server.

Security considerations for using REXEC server

These articles discuss methods for securing the REXEC server for authorized users and preventing access to the REXEC server.

The Remote EXECution server (REXEC) receives and runs commands from an REXEC client. A REXEC client is typically a PC or UNIX application that supports sending REXEC commands. The support that this server provides is similar to the capability that is available when you use the RCMD (Remote Command) sub-command for the FTP server.

Prevent REXEC access:

This article discusses the steps for preventing users from accessing the REXEC server.

If you do not want your system to accept commands from an REXEC client, do the following to prevent the REXEC server from running:

1. To prevent REXEC server jobs from starting automatically when you start TCP/IP, type the following: `CHGRXCA AUTOSTART(*NO)`

   AUTOSTART(*NO) is the default value.
2. To prevent someone from associating a user application, such as a socket application, with the port that the system normally uses for REXEC, do the following:
   a. Type `GO CFGTCP` to display the **Configure TCP/IP** menu.
   b. Select option **4** (Work with TCP/IP port restrictions).
   c. On the Work with TCP/IP Port Restrictions display, specify option **1** (Add).
   d. For the lower port range, specify `512`.
   e. For the upper port range, specify `*ONLY`.

      Note: The port restriction takes effect the next time that you start TCP/IP. If TCP/IP is active when you set the port restrictions, you should end TCP/IP and start it again.
   f. For the protocol, specify `*TCP`.

g. For the user profile field, specify a user profile name that is protected on your system. (A protected user profile is a user profile that does not own programs that adopt authority and does not have a password that is known by other users.) By restricting the port to a specific user, you automatically exclude all other users.

Secure the REXEC server:

This article provides recommendations for securing the REXEC server.

Following are considerations when you choose to run the Remote EXECution server on your system:

- An REXCD request includes a user ID, a password, and the command to run. Normal server authentication and authority checking applies:
  - The user profile and password combination must be valid.
  - The system enforces the Limit capabilities (LMTCPB) value for the user profile.
  - The user must be authorized to the command and to all of the resources that the command uses.
- The REXEC server provides exit points similar to the exit points that are available for the FTP server. You can use the Validation exit point to evaluate the command and decide whether to allow it.
- When you choose to run the REXEC server, you are running outside any menu access control that you have on your system. You must ensure that your object authority scheme is adequate to protect your resources.

Security considerations for using DNS server

These articles discuss methods for securing the DNS server for authorized users and preventing access to the DNS server.

Domain Name System (DNS) is a distributed database system for managing host names and their associated Internet Protocol (IP) addresses. On IBM Systems, the DNS server is intended to provide address translation for the internal, secure network (intranet). Using DNS means that people can use simple names, such as "www.ibm.com" to locate a host, rather than using the IP address (xxx.xxx.xxx.xxx).

Prevent DNS access:

This article discusses the steps for preventing users from accessing the DNS server.

If you do not want anyone to use the DNS server on your system, do the following:

1. To prevent DNS server jobs from starting automatically when you start TCP/IP, type the following: `CHGDNSA AUTOSTART(*NO)`

   AUTOSTART(*NO) is the default value.
2. To prevent someone from associating a user application, such as a socket application, with the port that the system normally uses for DNS, do the following:
   a. Type `GO CFGTCP` to display the **Configure TCP/IP** menu.
   b. Select option **4** (Work with TCP/IP port restrictions).
   c. On the Work with TCP/IP Port Restrictions display, specify option **1** (Add).
   d. For the lower port range, specify `53`.
   e. For the upper port range, specify `*ONLY`.

      Note: The port restriction takes effect the next time that you start TCP/IP. If TCP/IP is active when you set the port restrictions, you should end TCP/IP and start it again.
   f. For the protocol, specify `*TCP`.

g. For the user profile field, specify a user profile name that is protected on your system. (A protected user profile is a user profile that does not own programs that adopt authority and does not have a password that is known by other users.) By restricting the port to a specific user, you automatically exclude all other users.
h. Repeat steps 2c through 2g for the *UDP (user datagram) protocol.

Secure the DNS server:

This article provides recommendations for securing the DNS server.

Following are security considerations when you choose to run DNS on your system:

- The function that the DNS server provides is IP address translation and name translation. It does not provide any access to objects on your system. Your risk when an outsider accesses your DNS server is that the server provides an easy way to view the topology of your network. Your DNS might save a hacker some effort in determining the addresses of potential targets. However, your DNS does not provide information that will help to break into those target systems.
- Typically, you use the DNS server for your intranet. Therefore, you probably do not have a need to restrict the ability to query the DNS. However, you might, for example, have several subnetworks within your intranet. You might not want users from a different subnetwork to be able to query the DNS on your system. A security option of DNS lets you limit access to a primary domain. Use iSeries Navigator to specify IP addresses to which the DNS server should respond.

  Another security option lets you specify which secondary servers can copy information from your primary DNS server. When you use this option, your server will accept zone transfer requests (a request to copy information) only from the secondary servers that you explicitly list.
- Be sure to carefully restrict the ability to change the configuration file for your DNS server. Someone with malicious intent could, for example, change your DNS file to point to an IP address outside your network. They could simulate a server in your network and, perhaps, gain access to confidential information from users that visit the server.

Security considerations for using IBM HTTP server

These topics discuss methods for securing the IBM HTTP server for authorized users and preventing access to the HTTP server.

The HTTP server provides World Wide Web browser clients with access to system multimedia objects, such as HTML (Hypertext Markup Language) documents. It also supports the Common Gateway Interface (CGI) specification. Application programmers can write CGI programs to extend the functionality of the server.

The administrator can use Internet Connection Server or IBM HTTP server to run multiple servers concurrently on the same system. Each server that is running is called a server instance. Each server instance has a unique name. The administrator controls which instances are started and what each instance can do.

Important: You must have the *ADMIN instance of the HTTP server running when you use a Web browser to configure or administer any of the following:
- Firewall for iSeries
- Internet Connection Server
- Internet Connection Secure Server
- IBM HTTP Server

A user (Web site visitor) never sees a system Sign On display. However, the system administrator must explicitly authorize all HTML documents and CGI programs by defining them in HTTP directives. In addition, the administrator can set up both resource security and user authentication (user ID and password) for some or all requests.

An attack by a hacker could result in a denial of service to your Web server. Your server can detect a denial-of-service attack by measuring the timeout of certain clients' requests. If the server does not receive a request from the client, then your server determines that a denial-of-service attack is in progress. This occurs after making the initial client connection to your server. The server's default is to detect attacks.

Prevent HTTP access:

This article discusses the steps for preventing users from accessing the HTTP server.

If you do not want anyone to use the program to access your system, you should prevent the HTTP server from running. Do the following:

1. To prevent HTTP server jobs from starting automatically when you start TCP/IP, type the following:
   ```
   CHGHTTPA AUTOSTART(*NO)
   ```
   AUTOSTART(*NO) is the default value.
2. By default, the HTTP server job uses the QTMHHTTP user profile. To prevent the HTTP server from starting, set the status of the QTMHHTTP user profile to *DISABLED.

Control access to the HTTP server:

This article discusses considerations for protecting the contents of your Web site.

The primary purpose of running an HTTP server is to provide access for visitors to a Web site on your system. You might think of someone who visits your Web site as you would think of someone who views an advertisement in a trade journal. The visitor is not aware of the hardware and software running your Web site, such as the type of server you are using, and where your server is physically located. Usually, you do not want to put any barrier (such as a Sign On display) between a potential visitor and your Web site. However, you might want to restrict access to some of the documents or CGI programs that your Web site provides.

You might also want a single system to provide multiple logical Web sites. For example, your system might support different branches of your business that have different customer sets. For each of these branches of the business, you want a unique Web site that appears totally independent to the visitor. Additionally, you might want to provide internal Web sites (an intranet) with confidential information about your business.

As a security administrator, you need to protect the contents of your Web site while, at the same time, you need to ensure that your security practices do not negatively affect the value of your Web site. In addition, you need to ensure that HTTP activity does not jeopardize the integrity of your system or your network. The topics that follow provide security suggestions when you use the program.

Administration considerations:

This article provides recommendations for securing the Internet server.

Following are some security considerations for administering your Internet server.

- You perform setup and configuration functions by using a Web browser and the *ADMIN instance. For some functions, such as creating additional instances on the server, you must use the *ADMIN server.
- The default URL for the administration home page (the home page for the *ADMIN server) is published in the documentation for products that provide browser administration functions. Therefore, the default URL will probably be known by hackers and published in hacker forums, just like the default passwords for IBM-supplied user profiles are known and published. You can protect yourself from this exposure in several ways:
  - Only run the *ADMIN instance of the HTTP server when you need to perform administrative functions. Do not have the *ADMIN instance running all the time.

  - Activate SSL support for the *ADMIN instance (by using Digital Certificate Manager). The *ADMIN instance uses HTTP protection directives to require a user ID and password. When you use SSL, your user ID and password are encrypted (along with all the other information about your configuration that appears on the administration forms).
  - Use a firewall both to prevent access to the *ADMIN server from the Internet and to hide your system and domain names, which are part of the URL.
- When you perform administration functions, you must sign on with a user profile that has *IOSYSCFG special authority. You might also need authority to specific objects on the system, such as the following:
  - The libraries or directories that contain your HTML documents and CGI programs.
  - Any user profiles that you plan to swap to within the directives for the server.
  - The Access Control Lists (ACLs) for any directories that your directives use.
  - A validation list object for creating and maintaining user IDs and passwords.
- With both the *ADMIN server and TELNET, you have the capability to perform administration functions remotely, perhaps over an Internet connection. Be aware that if you perform administration over a public link (the Internet), you might be exposing a powerful user ID and password to sniffing. The ⌂sniffer⌂ can then use this user ID and password to attempt to access your system using, for example, TELNET or FTP.
- The HTTP directives provide the foundation for all activity on your server. The shipped configuration provides the capability to serve a default Welcome page. A client cannot view any documents except the Welcome page until the server administrator defines directives for the server. To define directives, use a Web browser and the *ADMIN server or the Work with HTTP Configuration (WRKHTTPCFG) command. Both methods require *IOSYSCFG special authority. When you connect your system to the Internet, it becomes even more critical to evaluate and control the number of users in your organization who have *IOSYSCFG special authority.

Notes:

1. TELNET, the Sign On display is treated like any other display. Although the password does not display when you type it, the system transmits it without any encryption or encoding.
2. With the *ADMIN server, the password is encoded not encrypted. The encoding scheme is an industry standard, and thus commonly known among the hacker community. Although the encoding is not easily understood by the casual ⌂sniffer,⌂ a sophisticated sniffer probably has tools to attempt to decode the password.

Security tip: If you plan to perform remote administration over the Internet, you should use the *ADMIN instance with SSL, so that your transmissions are encrypted. Do not use an insecure application. If you are using the *ADMIN server across an intranet of trusted users, you can safely use this for administration.

Protect resources:

The IBM HTTP server includes HTTP directives that can provide detailed control of the information assets that the server uses. You can use directives to control from which directories the Web server serves URLs for both HTML files and CGI programs, to swap to other user profiles, and to require authentication for some resources.

Following are some suggestions for using HTTP directives:

- The HTTP server starts from the basis of ⌂explicit authority.⌂ The server does not accept a request unless that request is explicitly defined in the directives. In other words, the server immediately rejects any request for a URL unless that URL is defined in the directives (either by name or generically).
- You can use protection directives to require a user ID and password before accepting a request for some or all of your resources.
  - When a user (client) requests a protected resource, the server challenges the browser for a user ID and password. The browser prompts the user to enter a user ID and password, and then sends the

information to the server. Some browsers store the user ID and password and send them automatically with subsequent requests. This frees the user from repeatedly entering the same user ID and password on each request.

Because some browsers store the user ID and password, you have the same user education task that you have when users enter your system through the system Sign On display or through a router. An unattended browser session represents a potential security exposure.

– You have three options for how the system handles user IDs and passwords (specified in the protection directives):

1. You can use normal system user profile and password validation. This is most commonly used to protect resources in an intranet (secure network).
2. You can create "Internet users": users that can be validated but do not have a user profile on the system. Internet users are implemented through a system object called a "validation list". Validation list objects contain lists of users and passwords that are specifically defined for use with a particular application.

   You decide how Internet user IDs and passwords are supplied (such as by an application, or by an administrator in response to an e-mail request), as well as how to manage Internet users. Use the HTTP server's browser-based interface to set this up.

   For nonsecure networks (the Internet), using Internet users provides better overall protection than using normal user profiles and passwords. The unique set of user IDs and passwords creates a built-in limitation on what those users can do. The user IDs and passwords are not available for normal sign-on (such as with TELNET or FTP). In addition, you are not exposing normal user IDs and passwords to sniffing.
3. Lightweight directory access protocol (LDAP) is a directory service protocol that provides access to a directory over a Transmission Control Protocol (TCP). It lets you store information in that directory service and query it. LDAP is now supported as a choice for user authentication.

   Notes:

   - When the browser sends the user ID and the password (whether for an user profile or an Internet user), they are encoded, not encrypted. The encoding scheme is an industry standard, and thus commonly known among the hacker community. Although the encoding is not easily understood by the casual "sniffer", a sophisticated sniffer probably has tools to attempt to decode them.
   - The system stores the validation object in a protected system area. You can access it only with defined system interfaces (APIs) and proper authorization.

– You can use Digital Certificate Manager (DCM) to create your own intranet Certificate Authority. Digital Certificate automatically associates a certificate with the owner's user profile. The certificate has the same authorizations and permissions as the associated profile.

- When the server accepts a request, normal system resource security takes over. The user profile that requests the resource must have authority to the resource (such as the folder or source physical file that contains the HTML document). By default, jobs run under the QTMHHTTP user profile. You can use a directive to swap to a different user profile. The system then uses that user profile's authority to access objects. Following are some considerations for this support:
  – Swapping user profiles can be particularly useful when your server provides more than one logical Web site. You can associate a different user profile with the directives for each Web site, and thus use normal system resource security to protect the documents for each site.
  – You can use the ability to swap user profiles in combination with the validation object. The server uses a unique user ID and password (separate from your normal user ID and password) to evaluate the initial request. After the server has authenticated the user, the system then swaps to a different user profile and thus takes advantage of resource security. The user is, thus, not aware of the true user profile name and cannot attempt to use it in other ways (such as FTP).
- Some HTTP server requests need to run a program on the HTTP server. For example, a program might access data on your system. Before the program can run, the server administrator must map the

request (URL) to a specific user-defined program that conforms to CGI user-interface standards. Following are some considerations for CGI programs:

- You can use the protection directives for CGI programs just as you do for HTML documents. Thus, you can require a user ID and password before running the program.
- By default, CGI programs run under the QTMHHTP1 user profile. You can swap to a different user profile before running the program. Therefore, you can set up normal system resource security for the resources that your CGI programs access.
- As security administrator, you should perform a security review before authorizing the use of any CGI program on your system. You should know where the program came from and what functions the CGI program performs. You should also monitor the capabilities of the user profiles under which you run CGI programs. You should also perform testing with CGI programs to determine, for example, whether you can gain access to a command line. Treat CGI programs with the same vigilance that you treat programs that adopt authority.
- In addition, be sure to evaluate what sensitive objects might have inappropriate public authority. A poorly designed CGI program might, in rare cases, allow a knowledgeable, devious user to attempt to roam your system.
- Use a specific user library, such as CGILIB, to hold all your CGI programs. Use object authority to control both who can place new objects in this library and who can run programs in this library. Use the directives to limit the HTTP server to running CGI programs that are in this library.

Tip: If your server provides multiple logical Web sites, you might want to set up a separate library for the CGI programs for each site.

Other security considerations

Following are additional security considerations:

- HTTP provides read-only access to your system. HTTP server requests cannot update or delete data on your system directly. However, you might have CGI programs that update data. Additionally, you can enable the Net.Data® CGI program to access your system database. The system uses a script (which is similar to an exit program) to evaluate requests to the Net.Data program. Therefore, the system administrator can control what actions the Net.Data program can take.
- The HTTP server provides an access log that you can use to monitor both accesses and attempted accesses through the server.

Security considerations for using SSL with HTTP server

IBM HTTP Server can provide secure Web connections to your system.

A secure web site means that transmissions between the client and the server (in both directions) are encrypted. These encrypted transmissions are safe both from the scrutiny of sniffers and from those who attempt either to capture or to alter the transmissions.

Note: Keep in mind that a secure Web site applies strictly to the security of the information that passes between client and server. The intent of this is not to reduce your server's vulnerability to hackers. However, it certainly limits the information that a would-be hacker can obtain easily through sniffing.

The topics on SSL and Webserving (HTTP) in the information center provides complete information for installing, configuring, and managing the encryption process. These topics provide both an overview of the server features and some considerations for using the server.

Internet Connection Server provides HTTP and HTTPS support when one of the following licensed programs is installed:

- 5722–NC1

When these options are installed, the product is referred to as the Internet Connection Secure Server.

Security that depends on encryption has several requirements:

- Both the sender and receiver (server and client) must understand the encryption mechanism and be able to perform encryption and decryption. The HTTP server requires an SSL-enabled client. (Most popular Web browsers are SSL-enabled.) The iSeries encryption licensed programs support several industry-standard encryption methods. When a client attempts to establish a secure session, the server and client negotiate to find the most secure encryption method that both of them support.
- The transmission must not be able to be decrypted by an eavesdropper. Thus, encryption methods require both parties to have an encryption/decryption private key that only they know. If you want to have a secure external Web site, you should use an independent certificate authority (CA) to create and issue digital certificates to users and servers. The certificate authority is known as a trusted party.

Encryption protects the confidentiality of transmitted information. However, for sensitive information, such as financial information, you want integrity and authenticity in addition to confidentiality. The client and (optionally) the server must trust the party on the other end (through an independent reference) and they must be sure that the transmission has not been altered. The digital signature that is provided by a certification authority (CA) provides these assurances of authenticity and integrity. The SSL protocol provides authentication by verifying the digital signature of the server's certificate (and optionally the client's certificate).

Encryption and decryption require processing time and will affect the performance of your transmissions. Therefore, iSeries servers provide the capability to run both the programs for secure and insecure serving at the same time. You can use the insecure HTTP server to serve documents that do not require security, such as your product catalog. These documents will have a URL that starts with http://. You can use a secure HTTP server for sensitive information such as the form where the customer enters credit card information. The program can serve documents whose URL starts either with http:// or with https://.

Reminder: It is good Internet etiquette to inform your clients when transmissions are secure and not secure, particularly when your Web site only uses a secure server for some documents.

Keep in mind that encryption requires both a secure client and a secure server. Secure browsers (HTTP clients) have become fairly common.

Security considerations for LDAP

Lightweight Directory Access Protocol (LDAP) security features include Secure Sockets Layer (SSL), Acess Control Lists, and CRAM-MD5 password encryption.

In V5R1, Kerberos connections and Security auditing support were added to enhance LDAP security. See Directory Services (LDAP) or more information on these topics.

Security considerations for LPD

LPD (line printer daemon) provides the capability to distribute printer output to your system. The system does not perform any sign-on processing for LPD.

Prevent LPD access:

These instructions explain how to prevent LDP access.

If you do not want anyone to use LPD to access your system, you should prevent the LPD server from running. Do the following:

1. To prevent LPD server jobs from starting automatically when you start TCP/IP, type the following:
   `CHGLPDA AUTOSTART(*NO)`

   Note:

a. AUTOSTART(*YES) is the default value.
b. Control which TCP/IP servers start automatically provides more information about controlling which TCP/IP servers start automatically.

2. To prevent someone from associating a user application, such as a socket application, with the port that the system normally uses for LPD, do the following:
   a. Type GO CFGTCP to display the Configure TCP/IP menu.
   b. Select option 4 (Work with TCP/IP port restrictions).
   c. On the Work with TCP/IP Port Restrictions display, specify option 1 (Add).
   d. For the lower port range, specify 515.
   e. For the upper port range, specify *ONLY.

   Note:
   a. The port restriction takes effect the next time that you start TCP/IP. If TCP/IP is active when you set the port restrictions, you should end TCP/IP and start it again.
   b. RFC1700 provides information about common port number assignments.
3. For the protocol, specify *TCP.
4. For the user profile field, specify a user profile name that is protected on your system. A protected user profile is a user profile that does not own programs that adopt authority and does not have a password that is known by other users. By restricting the port to a specific user, you automatically exclude all other users.
5. Repeat steps 2c through 2g for the *UDP protocol.

Control LPD access:

If you want to allow LPD clients to access your system, be aware of the following security issues.

These security issues are important for you to be aware of:

- To prevent a user from swamping your system with unwanted objects, be sure that you have set adequate threshold limits for your auxiliary storage pools (ASPs). You can display and set the thresholds for ASPs by using either system service tools (SST) or dedicated service tools (DST). The Backup and Recovery book provides more information about ASP thresholds.
- You can use the authority to output queues to restrict who can send spooled files to your system. LPD users without a user ID use the QTMPLPD user profile. You can give this user profile access to only a few output queues.

Security considerations for SNMP

SNMP provides a means for managing the gateways, routers, and hosts in a network environment.

The system can act as a simple network management protocol (SNMP) agent in a network. An SNMP agent gathers information about the system and performs functions that remote SNMP network managers request.

Prevent SNMP access:

You can follow these instructions and prevent SNMP access to your system.

If you do not want anyone to use SNMP to access your system, you should prevent the SNMP server from running. Do the following:

1. To prevent SNMP server jobs from starting automatically when you start TCP/IP, type the following:
   `CHGSNMPA AUTOSTART(*NO)`

   Note:

a. AUTOSTART(*YES) is the default value.
b. Control which TCP/IP servers start automatically provides more information about controlling which TCP/IP servers start automatically.

2. To prevent someone from associating a user application, such as a socket application, with the port that the system normally uses for SNMP, do the following:
   a. Type GO CFGTCP to display the Configure TCP/IP menu.
   b. Select option 4 (Work with TCP/IP port restrictions).
   c. On the Work with TCP/IP Port Restrictions display, specify option 1 (Add).
   d. For the lower port range, specify 161.
   e. For the upper port range, specify *ONLY.

   Note:
   a. The port restriction takes effect the next time that you start TCP/IP. If TCP/IP is active when you set the port restrictions, you should end TCP/IP and start it again.
   b. RFC1700 provides information about common port number assignments.
3. For the protocol, specify *TCP.
4. For the user profile field, specify a user profile name that is protected on your system. A protected user profile is a user profile that does not own programs that adopt authority and does not have a password that is known by other users. By restricting the port to a specific user, you automatically exclude all other users.
5. Repeat steps 2c through 2g for the *UDP protocol.

Control SNMP access:

If you want to allow SNMP managers to access your system, you need to be aware of the following security issues.

These security issues are important to be aware of:

- Someone who can access your network with SNMP can gather information about your network. Information that you have hidden by using aliases and a domain-name server becomes available to the would-be intruder through SNMP. Additionally, an intruder might use SNMP to alter your network configuration and disrupt your communications.
- SNMP relies on a community name for access. Conceptually, the community name is similar to a password. The community name is not encrypted. Therefore, it is vulnerable to sniffing. Use the Add Community for SNMP (ADDCOMSNMP) command to set the manager internet address (INTNETADR) parameter to one or more specific IP addresses instead of *ANY. You can also set the OBJACC parameter of the ADDCOMSNMP or CHGCOMSNMP commands to *NONE to prevent the managers in a community from accessing any MIB objects. This is intended to just be done temporarily to deny access to managers in a community without removing the community.

Security considerations for INETD server

Unlike most TCP/IP servers, the INETD server does not provide one single service to clients.

The INETD server provides a variety of miscellaneous services that administrators can customize. For that reason, the INETD server is sometimes called ⌂the super server⌂. The INETD server has the following built-in services:

- Time
- Daytime
- Echo
- Discard
- Changed

These services are supported for both TCP and UDP. For UDP, the echo, time, daytime, and changed services receive UDP packets, then send the packets back to the originator. The echo server echoes back packets that it receives, the time and daytime servers generate the time in a specific format and sends it back, and the changed server generates a packet of printable ASCII characters and sends it back.

The nature of these UDP services makes a system vulnerable to a denial of service attack. For example, assume that you have two iSeries servers: SYSTEMA and SYSTEMB. A malicious programmer could forge the IP header and the UDP header with a source address of SYSTEMA and a UDP port number of the time server. He can then send that packet to the time server on SYSTEMB, which will send the time to SYSTEMA, which will respond back to SYSTEMB, and so on, generating a continuous loop and consuming CPU resources on both systems, as well as network bandwidth.

Therefore, you should consider the risk of such an attack on your iSeries system, and only run these services on a secure network. The INETD server is shipped to not be auto started when you start TCP/IP. You can configure whether or not to start the services when INETD is started. By default, the TCP and UDP time servers and daytime servers are both started when you start the INETD server.

There are two configuration files for the INETD server: `/QIBM/UserData/OS400/inetd/inetd.conf` `/QIBM/ProdData/OS400/inetd/inetd.conf`

These files determine what programs start when the INETD server starts. They also determine what user profile these programs are running under when INETD starts them.

Note: The configuration file in proddata should never be modified. It is replaced each time the system is reloaded. Customer configuration changes should only be placed in the file, in the UserData directory tree, as this file in not updated during release upgrades.

If a malicious programmer got access to these files, she could configure them to start any program when INETD started. Therefore it is very important to protect these files. By default they require QSECOFR authority to make changes. You should not reduce the authority required to access them.

Note: Do not modify the configuration file in the ProdData directory. That file is replaced each time that the system is reloaded. Customer configuration changes should only be placed in the file in the UserData directory tree, as that file is not updated during release upgrades.

Security considerations for limiting TCP/IP roaming

If your system is connected to a network, you may want to limit your users' ability to roam the network with TCP/IP applications.

One way to do this is to restrict access to the following client TCP/IP commands:

Note: These commands might exist in several libraries on your system. They are in both the QSYS library and the QTCP library, at a minimum. Be sure to locate and secure all occurrences.

- STRTCPFTP
- FTP
- STRTCPTELN
- TELNET
- LPR
- SNDTCPSPLF
- RUNRMTCMD (REXEC client)

Your users' possible destinations are determined by the following:

- Entries in your TCP/IP host table.

- *DFTROUTE entry in the TCP/IP route table. This allows users to enter the IP address of the next-hop system when their destination is an unknown network. A user can reach or contact a remote network by using the default route.
- Remote name server configuration. This support allows another server in the network to locate host names for your users.
- Remote system table.

You need to control who can add entries to these tables and change your configuration. You also need to understand the implications of your table entries and your configuration.

Be aware that a knowledgeable user with access to an ILE C compiler can create a socket program that can attach to a TCP or UDP port. You can make this more difficult by restricting access to the following sockets interface files in the QSYSINC library:

- SYS
- NETINET
- H
- ARPA
- Sockets and SSL

For service programs, you can restrict use of socket and SSL applications that are already compiled by restricting use of these service programs:

- QSOSRV1
- QSOSRV2
- QSOSKIT(SSL)
- QSOSSLSR(SSL)

The service programs are shipped with public authority *USE, but the authority can be changed to *EXCLUDE (or another value as needed).

Security considerations for using RouteD

This topic discusses security considerations for using the Route Daemon (RouteD) server. RouteD, provides support for the Routing Information Protocol on the system.

The Route Daemon (RouteD) server provides support for the Routing Information Protocol (RIP) on IBM Systems. RIP is the most widely used of routing protocols. It is an Interior Gateway Protocol that assists TCP/IP in the routing of IP packets within an autonomous system.

RouteD is intended to increase the efficiency of network traffic by allowing systems within a trusted network to update each other with current route information. When you run RouteD, your system can receive updates from other participating systems about how transmissions (packets) should be routed. Therefore, if your RouteD server is accessible to a hacker, the hacker might use it to reroute your packets through a system that can sniff or modify those packets. Following are suggestions for RouteD security:

- IBM Systems use RIPv1, which does not provide any method for authenticating routers. It is intended for use within a trusted network. If your system is in a network with other systems that you do not “trust,” you should not run the RouteD server. To ensure that the RouteD server does not start automatically, type the following: `CHGRTDA AUTOSTART(*NO)`
- Make sure that you control who can change the RouteD configuration, which requires *IOSYSCFG special authority.
- If your system participates in more than one network (for example, an intranet and the Internet), you can configure the RouteD server to send and accept updates only with the secure network.

Manage security

Once you've planned and implemented your security strategy, there remains the task of managing the security of your system.

These topics will guide you through setting up your security management plan:

- Back up and recover security information
- Manage security information
- Manage service tools user IDs
- Protect against computer viruses

Restrict save and restore capability

Part of your security system should be controlling users' save and restore capabilities.

Most users do not need to save and restore objects on your system. The save commands provide the possibility of copying important assets of your organization to media or to another system. Most save commands support save files that can be sent to another system (by using the SNDNETF file command) without having access to media or a save/restore device.

Restore commands provide the opportunity to restore unauthorized objects, such as programs, commands, and files, to your system. You can also restore information without access to media or to a save/restore device by using save files. Save files can be sent from another system by using the SNDNETF command or by using the FTP function.

Following are suggestions for restricting save and restore operations on your system:

- Control which users have *SAVSYS special authority. *SAVSYS special authority allows the user to save and restore objects even when the user does not have the necessary authority to the objects.
- Control physical access to save and restore devices.
- Restrict access to the save and restore commands. When you install i5/OS licensed programs, the public authority for the RSTxxx commands is *EXCLUDE. Public authority for the SAVxxx commands is *USE. Consider changing the public authority for SAVxxx commands to *EXCLUDE. Carefully limit the users that you authorize to the RSTxxx commands.
- Use the QALWOBJRST system value to restrict restoration of system-state programs, programs that adopt authority, and objects that have validation errors.
- Use the QVFYOBJRST system value to control restoring signed objects on your system.
- Use the QFRCCVNRST system value to control the recreation of certain objects being restored on your system.
- Use security auditing to monitor restore operations. Include *SAVRST in the QAUDLVL system value, and periodically print audit records that are created by restore operations.

Save security information

This topic presents an overview of how you save and restore security information.

When you plan the backup and recovery of your system, you need to consider the security of your information as well as the information itself. See the Information Center topic, Backup, Recovery, and Availability to help you design a complete backup and recovery plan. The following topics describe how you back up and restore the security information that you create when you set up security:

Related concepts

"User security" on page 12
From a user's point of view, security affects how they use and complete tasks on the system.

Save system values

This article describes the task, save system values, explains why it is important, and provides step-by-step instructions.

Saving system values System values are stored in the system library, QSYS. You save the QSYS library when you do the following:

- Use the Save System (SAVSYS) command.
- Use the option to save the entire system from the Save menu.
- Use the option to save system information from the Save menu.
- Use the option to back up the entire system from the Run Backup (RUNBCKUP) menu.

If you need to recover your entire system, you automatically restore your system values when you restore your operating system. See ″Saving group and user profiles″ next.

Another option for saving System Values in V5R4 is the SAVSYSINF command.

Save group and user profiles

Group and user profiles are stored in the QSYS library. You save them when you use the Save System (SAVSYS) command or select the menu option to save the entire system.

You can also save group and user profiles by using the Save Security Data (SAVSECDTA) command. Restore user profiles by using the Restore User Profile (RSTUSRPRF) command. The normal sequence follows:

1. Restore the operating system, which restores library QSYS.
2. Restore user profiles.
3. Restore the remaining libraries.
4. Restore authority to objects using the
5. Restore Authority (RSTAUT) command.

Save job descriptions

When you create a job description, you specify a library where it should reside. IBM recommends creating job descriptions into the QGPL library.

You can save job descriptions by saving the library in which they reside. Use the Save Library (SAVLIB) command to do this. You can also save a job description by using the Save Object (SAVOBJ) command.

You can restore the contents of a library by using the Restore Library (RSTLIB) command. You can restore an individual job description by using the Restore Object (RSTOBJ) command.

Save resource security information

Resource security, which defines how users can work with objects, consists of different types of information that is stored in several different places:

Type of information	Where it is stored	How it is saved	How it is restored
Public authority	With the object	SAVxxx command[1]	RSTxxx command[2]
Object auditing value	With the object	SAVxxx command[1]	RSTxxx command[2]
Object ownership	With the object	SAVxxx command[1]	RSTxxx command[2]
Primary group	With the object	SAVxxx command[1]	RSTxxx command[2]
Authorization list	QSYS library	SAVESYS or SAVSECDTA	RSTUSRPRF, USRPRF (*ALL)
Link between object and authorization list	With the object	SAVxxx command[1]	RSTxxx command[2]

Type of information	Where it is stored	How it is saved	How it is restored
Private authority	With the user profile	SAVESYS or SAVSECDTA	RSTAUT

[1] You can save most object types by using the SAVOBJ or SAVLIB commands. Some object types, such as configurations, have a special save command.

[2] You can restore most object types by using the RSTOBJ or RSTLIB commands. Some object types, such as configurations, have a special restore command.

When you need to recover an application or your entire system, you need to plan the steps carefully, including recovery of the authority to objects. Following are the basic steps necessary to recover the resource security information for an application:

1. If necessary, restore user profiles, including the profiles which own the application. You can restore specific profiles or all profiles with the RSTUSRPRF command.
2. Restore any authorization lists that are used by the application. You restore authorization lists when you use RSTUSRPRF USRPRF(*ALL).

   Note: This restores all the user profile values, including passwords, from the backup media.
3. Restore the application libraries by using the RSTLIB or RSTOBJ command. This recovers object ownership, public authority, and the links between objects and authorization lists.
4. Restore private authority to objects by using the RSTAUT command. The RSTAUT command also restores user authorities to authorization lists. You can restore authority for specific users or all users.

Save the default owner profile (QDFTOWN)

If you restore an object and the owner profile is not on the system, the system transfers ownership of the object to a default profile that is called QDFTOWN.

Once you recover the owner profile or create it again, you can transfer ownership back by using the Work with Object by Owner (WRKOBJOWN) command.

Restore security information

Recovering your system often requires restoring data and associated security information.

The usual sequence for recovery is:

1. Restore user profiles and authorization lists (RSTUSRPRF USRPRF(*ALL)).
2. Restore objects (RSTLIB, RSTOBJ, or RSTCFG).
3. Restore the private authorities to objects (RSTAUT).

Restore related system values

This information enables you to control how and which security-related objects are restored on the system.

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

Authority:
*ALLOBJ and *SECADM

Journal Entry:
SV

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

Following are descriptions of system values that relate to restoring security-related objects on the system which should be considered when restoring objects as well.

QVFYOBJRST
Verify object on restore

QFRCCVNRST
Force conversion on restore

QALWOBJRST
Allow restoring of security sensitive objects

Restore user profiles

Restore user profiles

Some changes may be made to a user profile when it is restored.

The following applies: If profiles are being restored individually (RSTUSRPRF USRPRF(*ALL) is not specified), SECDTA(*PWDGRP) is not requested, and the profile being restored does not exist on the system, these fields are changed to *NONE:

- Group profile name (GRPPRF)
- Password (PASSWORD)
- Document password (DOCPWD)
- Supplemental group profiles (SUPGRPPRF)

Product passwords are changed to *NONE, so they will be incorrect after restoring an individual user profile that did not exist on the system.

- If profiles are being restored individually (RSTUSRPRF USRPRF(*ALL) is not specified) SECDTA(*PWDGRP) is not requested, and the profile exists on the system, the password, document password, and group profile are not changed. User profiles can be restored individually with the password and group information restored from the save media by specifying the SECDTA(*PWDGRP) parameter on the RSTUSRPRF command. *ALLOBJ and *SECADM special authorities are required to restore the password and group information when restoring individual profiles. Product passwords restored with the user profile will be incorrect after restoring an individual user profile that existed on the system, unless the SECDTA(*PWDGRP) parameter is specified on the RSTUSRPRF command.
- If all user profiles are being restored to your system, all the fields in any profiles that already exist on the system are restored from the save media, including the password.

  Note:

  1. User Profiles saved from a system with a different password level (QPWDLVL system value) than the system that is being restored may result in having a password that is not valid on the restored system. For example, if the saved user profile came from a system that was running password level 2, the user could have a password of ⌂This is my password⌂. This password would not be valid on a system running password level 0 or 1.
  2. Keep a record of the security officer (QSECOFR) password associated with each version of your security information that is saved to make sure you can sign on to your system if you need to do a complete restore operation.

  You can use DST (Dedicated Service Tools) to reset the password for the QSECOFR profile. See Service tools topic in the Information Center for instructions. See “Prerequisite and related information” on page xvi for more information about accessing the Information Center.
- If a profile exists on the system, the restore operation does not change the uid or gid.
- If a profile does not exist on the system, the uid and gid for a profile are restored from the save media. If either the uid or the gid already exists on the system, the system generates a new value and issues a message (CPI3810).
- *ALLOBJ special authority is removed from user profiles being restored to a system at security level 30 or higher in either of these situations:

– The profile was saved from a different system and the user performing the RSTUSRPRF does not have *ALLOBJ and *SECADM special authorities.
– The profile was saved from the same system at security level 10 or 20.

Note: The system uses the machine serial number on the system and on the save media to determine whether objects are being restored to the same system or a different system.

*ALLOBJ special authority is not removed from these IBM-supplied profiles:

– QSYS (system) user profile
– QSECOFR (security officer) user profile
– QLPAUTO (licensed program automatic install) user profile
– QLPINSTALL (licensed program install) user profile

Restore objects

When you restore an object to the system, the system uses the authority information stored with the object.

The following applies to security of the restored object:

Object ownership:

- If the profile that owns the object is on the system, ownership is restored to that profile.
- If the owner profile does not exist on the system, ownership of the object is given to the QDFTOWN (default owner) user profile.
- If the object exists on the system and the owner on the system is different from the owner on the save media, the object is not restored unless ALWOBJDIF(*ALL) is specified. In that case, the object is restored and the owner on the system is used.

Primary group:

For an object that does not exist on the system:

- If the profile that is the primary group for the object is on the system, the primary group value and authority are restored for the object.
- If the profile that is the primary group does not exist on the system:
  – The primary group for the object is set to none.
  – The primary group authority is set to no authority.

When an existing object is restored, the primary group for the object is not changed by the restore operation.

Public authority:

- If the object being restored does not exist on the system, public authority is set to the public authority of the saved object.
- If the object being restored does exist and is being replaced, public authority is not changed. The public authority from the saved version of the object is not used.
- The CRTAUT for the library is not used when restoring objects to the library.

Authorization list:

- If an object, other than a document or folder, already exists on the system and is linked to an authorization list, the ALWOBJDIF parameter determines the result:
  – If ALWOBJDIF(*NONE) is specified, the existing object must have the same authorization list as the saved object. If not, the object is not restored. –
  – If ALWOBJDIF(*ALL) is specified, the object is restored. The object is linked to the authorization list associated with the existing object.

- If a document or folder that already exists on the system is restored, the authorization list associated with the object on the system is used. The authorization list from the saved document or folder is not used.
- If the authorization list does not exist on the system, the object is restored without being linked to an authorization list and the public authority is changed to *EXCLUDE.
- If the object is being restored on the same system from which it was saved, the object is linked to the authorization list again.
- If the object is being restored on a different system, the ALWOBJDIF parameter on the restore command is used to determine whether the object is linked to the authorization list:
  - If ALWOBJDIF(*ALL) is specified, the object is linked to the authorization list.
  - If ALWOBJDIF(*NONE) is specified, then the object is not linked to the authorization list and the public authority of the object is changed to *EXCLUDE.

Private authorities:
- Private authority is saved with user profiles, not with objects.
- If user profiles have private authority to an object being restored, those private authorities are usually not affected. Restoring certain types of programs may result in private authorities being revoked.
- If an object is deleted from the system and then restored from a saved version, private authority for the object no longer exists on the system. When an object is deleted, all private authority to the object is removed from user profiles.
- If private authorities need to be recovered, the Restore Authority (RSTAUT) command must be used. The normal sequence is:
  1. Restore user profiles
  2. Restore objects
  3. Restore authority

Object Auditing:
- If the object being restored does not exist on the system, the object auditing (OBJAUD) value of the saved object is restored.
- If the object being restored does exist and is being replaced, the object auditing value is not changed. The OBJAUD value of the saved version of the object is not restored.
- If a library being restored does not exist on the system, the create object auditing (CRTOBJAUD) value for the library is restored.
- If a library being restored exists and is being replaced, the CRTOBJAUD value for the library is not restored. The CRTOBJAUD value for the existing library is used.

Authority Holder:
- If a file is restored and an authority holder exists for that file name and the library to which it is being restored, the file is linked to the authority holder.
- The authority information associated with the authority holder replaces the public authority and owner information saved with the file.

Domain Objects: For systems running Version 2 Release 3 or later of the OS/400 licensed program, the system restricts user domain objects (*USRSPC, *USRIDX, and *USRQ) to the libraries specified in the QALWUSRDMN system value. If a library is removed from the QALWUSRDMN system value after a user domain object of type *USRSPC, *USRIDX, or *USRQ is saved, the system changes the object to system domain when it is restored.

Function Registration Information: The function registration information can be restored by restoring the QUSEXRGOBJ *EXITRG object into QUSRSYS. This restores all of the registered functions. The usage information associated with the functions is restored when user profiles and authorities are restored.

Applications that Use Certificates Registration: The applications that use certificates registration information can be restored by restoring the QUSEXRGOBJ *EXITRG object into QUSRSYS. This restores all of the registered applications. The association of the application to its certificate information can be restored by restoring the QYCDCERTI *USRIDX object into QUSRSYS.

Refer to Restore authority for more information.

Restore authority

When security information is restored, private authorities must be rebuilt. When you restore a user profile that has an authority table, the authority table for the profile is also restored. The Restore Authority (RSTAUT) command rebuilds the private authority in the user profile using the information from the authority table.

The grant authority operation is run for each private authority in the authority table. If authority is being restored for many profiles and many private authorities exist in the authority tables, this can be a lengthy process. The RSTUSRPRF and RSTAUT commands can be run for a single profile, a list of profiles, a generic profile name, or all profiles. The system searches the save media or save file created by the SAVSECDTA or SAVSYS command or the QSRSAVO API to find the profiles you want to restore.

Restoring Field Authority:

The following steps are required to restore private field authorities for database files that do not already exist on the system:

- Restore or create the necessary user profiles.
- Restore the files.
- Run the Restore Authority (RSTAUT) command.

The private field authorities are not fully restored until the private object authorities that they restrict are also established again.

Refer to Restore programs for more information.

Restore programs

Restoring programs to your system that are obtained from an unknown source poses a security exposure. Programs might perform operations that break your security requirements. Of particular concern are programs that contain restricted instructions, programs that adopt their owner authority, and programs that have been tampered with.

This includes object types *PGM, *SRVPGM, *MODULE, and *CRQD. You can use the QVFYOBJRST, QFRCCVNRST, and QALWOBJRST system values to prevent these object types from being restored to your system. See Security-Related Restore System Values for more information about these system values.

The system uses a validation value to help protect programs. This value is stored with a program and recalculated when the program is restored. The system's actions are determined by the ALWOBJDIF parameter on the restore command and the force conversion on restore (QFRCCVNRST) system value.

Note: Programs that are created for systems running Version 5 Release 1 or later versions of OS/400 or i5/OS contain information that allows the program to be re-created at restore time if necessary. The information needed to re-create the program remains with the program even when the observability of the program is removed. If a program validation error is determined to exist at the time the program is restored, the program will be re-created in order to correct the program validation error. The action of re-creating the program at restore time is not new to iseries Version 5 Release 1. In previous releases, any program validation error that was encountered at restore time resulted in the program being re-created if possible (if observability existed in the program

being restored). The difference with Version 5 Release 1 or later versions of programs is that the information needed to re-create these programs remain, even when observability is removed from the program.

Restoring Programs That Adopt the Owner's Authority:

When a program is restored that adopts owner authority, the ownership and authority to the program may be changed. The following applies:

- The user profile doing the restore operation must either own the program or have *ALLOBJ and *SECADM special authorities.
- The user profile doing the restore operation can receive the authority to restore the program by
  - Being the program owner.
  - Being a member of the group profile that owns the program (unless you have private authority to the program).
  - Having *ALLOBJ and *SECADM special authority.
  - Being a member of a group profile that has *ALLOBJ and *SECADM special authority.
  - Running under adopted authority that meets one of the tests just listed.
- If the restoring profile does not have adequate authority, all public and private authorities to the program are revoked, and the public authority is changed to *EXCLUDE.
- If the owner of the program does not exist on the system, ownership is given to the QDFTOWN user profile. Public authority is changed to *EXCLUDE and the authorization list is removed.

Refer to Restore licensed programs for more information.

Restore licensed programs

The Restore Licensed Programs (RSTLICPGM) command is used to install IBM-supplied programs on your system. It can also be used to install non-IBM programs created using the SystemView* System Manager/400* licensed program.

When your system is shipped, only users with *ALLOBJ special authority can use the RSTLICPGM command. The RSTLICPGM procedure calls an exit program to install programs that are not supplied by IBM.

To protect security on your system, the exit program should not run using a profile with *ALLOBJ special authority. Use a program that adopts *ALLOBJ special authority to run the RSTLICPGM command, instead of having a user with *ALLOBJ authority run the command directly.

Following is an example of this technique. The program to be installed using the RSTLICPGM command is called CPAPP (Contracts and Pricing).

1. Create a user profile with sufficient authority to successfully install the application. Do not give this profile *ALLOBJ special authority. For the example, the user profile is called OWNCP.
2. Write a program to install the application. For the example, the program is called CPINST: PGM RSTLICPGM CPAPP ENDPGM
3. Create the CPINST program to adopt the authority of a user with *ALLOBJ special authority, such as QSECOFR, and authorize OWNCP to the program:

   ```
   CRTCLPGM QGPL/CPINST USRPRF(*OWNER) +
   AUT(*EXCLUDE) GRTOBJAUT OBJ(CPINST) OBJTYP(*PGM) +
   USER(OWNCP) AUT(*USE)
   ```

4. Sign on as OWNCP and call the CPINST program. When the CPINST program runs the RSTLICPGM command, you are running under QSECOFR authority. When the exit program runs to install the CPAPP programs, it drops adopted authority. The programs called by the exit program run under the authority of OWNCP.

For procedural steps, refer to.Restore authorization lists.

Restore authorization lists

Authorization lists are saved by either the SAVSECDTA command or the SAVSYS command.

Authorization lists are restored by the command: RSTUSRPRF USRPRF(*ALL) No method exists for restoring an individual authorization list. When you restore an authorization list, authority and ownership are established just as they are for any other object that is restored.

The link between authorization lists and objects is established if the objects are restored after the authorization list. Users' private authorities to the list are restored using the RSTAUT command.

Next, restore the operating system.

Restore the operating system

When you perform a manual IPL on your system, the IPL or Install the System menu provides an option to install the operating system.

The dedicated service tools (DST) function provides the ability to require anyone using this menu option to enter the DST security password. You can use this to prevent someone from restoring an unauthorized copy of the operating system. To secure the installation of your operating system, do the following:

1. Perform a manual IPL.
2. From the IPL or Install the System menu, select DST.
3. From the Use DST menu, select the option to work with the DST environment.
4. Select the option to change DST passwords.
5. Select the option to change the operating system install security.
6. Specify 1 (secure).
7. Press F3 (exit) until you return to the IPL or Install the System menu.
8. Complete the manual IPL and return the keylock to its normal position.

Note:

1. If you no longer want to secure the installation of the operating system, follow the same steps and specify 2 (not secure).
2. You can also prevent installation of the operating system by keeping your keylock switch in the normal position and removing the key.

Refer to Manage security information for more information.

Manage security information

This article describes the tasks for managing security information.

Now that you have planned the security for your system, you need to ensure that your plan remains effective as your business needs change. This topic emphasizes simplicity as an essential goal in designing security. You have designed user groups as patterns for individual users. You have tried to use public authority, authorization lists, and library authority rather than specific individual authorities. Take advantage of that approach as you manage security:

- When you add a new user group or a new application, use the techniques that you used to plan security.
- When you need to make changes to security, try to take a general approach rather than creating an exception to solve a specific problem.

Work with security commands

This article describes how use security commands to display, change, and delete security information.

The table below shows what commands you use to work with security objects on the system. You can use these commands to perform these tasks:

- View and list security information.
- Change security information.
- Delete security information.

Table 111. Security Commands

Security Object	How to View	How to Change	How to Delete
System Value	WRKSYSVAL DSPSYSVAL	WRKSYSVAL CHGSYSVAL	Cannot be deleted
Job Description	WRKJOBD DSPJOBD	WRKJOBD CHGJOBD	DLTJOBD
Group Profile	WRKUSRPRF DSPUSRPRF DSPAUTUSR	WRKUSRPRF CHGUSRPRF	DLTUSRPRF[1], [2]
User Profile	WRKUSRPRF DSPUSRPRF DSPAUTUSR	WRKUSRPRF CHGUSRPRF CHGUSRAUD	DLTUSRPRF[1]
Object Authorities	DSPAUT DSPOBJAUT DSPUSRPRF TYPE(*OBJAUT)	CHGAUT EDTOBJAUT GRTOBJAUT WRKAUT	EDTOBJAUT RVKOBJAUT WRKAUT
Object Ownership	WRKOBJOWN DSPOBJAUT DSPUSRPRF TYPE(*OBJOWN)	CHGOBJOWN CHGOWN	CHGOBJOWN CHGOWN allows you to revoke the rights of the previous owner.
Primary Group	DSPOBJAUT WRKOBJPGP DSPUSRPRF TYPE(*OBJPGP)	CHGOBJPGP CHGPGP	CHGOBJPGP CHGPGP set primary group to *NONE
Object Auditing	DSPOBJD	CHGOBJAUD CHGAUD	CHGOBJAUD (set to *NONE) CHGAUD
Authorization List	DSPAUTL DSPAUTLOBJ	EDTAUTL (user authority to a list) EDTOBJAUT (object secured by list) ADDAUTLE CHGAUTLE GRTOBJAUT	DLTAUTL (entire list)[3] RMVAUTLE (remove user authority to the list) EDTOBJAUT (object secured by list) RVKOBJAUT

1. IBM recommends using the remove option from the Work with User Enrollment display for deleting a profile. Using this option, you can delete any objects that are owned by the profile or reassign them to a new owner. Certain DLTUSRPRF command parameters allow you to delete all objects that are owned by the user or assign them all to a new owner. You cannot delete a profile unless you delete or reassign owned objects. You also cannot delete a profile that is the primary group for any objects.
2. You cannot delete a group profile that has any members. Use the *GRPMBR option of the DSPUSRPRF command to list the members of the group. Change the Group Profile field in each of the individual group profiles before deleting the group profile.
3. You cannot delete an authorization list that is used to secure objects. Use the DSPAUTLOBJ command to list the objects that are secured by the list. Change the authority of any objects that are secured by the list by using the EDTOBJAUT command.

Viewing and listing security information

You can list security information by using a display (DSP) command with a print (*PRINT) option. For example, to display an authorization list called MYLIST, type `DSPAUTL MYLIST *PRINT`.

Some display commands provide options for different types of lists. For example, when you created individual user profiles, you used the *GRPMBR option of the DSPUSRPRF command to list all the members of a group profile. Use prompting (F4) and online information to find out what lists are available for security objects.

You can use the Display commands to view security information at your display station. You can also use the Work with... (WRK) commands, which provide more function. The Work With... commands give you a list display. You can use this display to change, delete, and view information.

You can also use security commands to list or view information by using a generic name. If you type `WRKUSRPRF DPT*`, your Work with User Enrollment display or Work with User Profile display shows only profiles that start with the characters *DPT*. Use online information for a command to find out which parameters allow generic names.

Changing security information

You can change security information interactively by using a Work With... (WRK) or Edit... (EDT) command. You can view the information, change it, and view the information again after the change.

You can also change security information without viewing it before and after the change by using a Change... (CHG) or Grant... (GRT) command. This method is particularly useful for making a change to more than one object at a time. For example, you used the GRTOBJAUT command to set public authority for all the objects in a library.

Deleting security information

You can delete or remove certain types of security information interactively by using the Work with... (WRK) or Edit... (EDT) commands. You can also use Delete... (DLT), Remove... (RMV), and Revoke... (RVK) commands to delete security information. Often, you must meet certain conditions before the system allows you to delete security information.

Add a new user to the system

This information explains how to add new users to the system.

You might need to create new user groups for several reasons:

- Additional departments need to use the system.
- You discover that you need to make user groups more specific to meet your resource security needs.
- Your company reorganized some departments.

When you need to add a new user to the system, use the following procedure:

1. Assign the person to a user group. Use the User Group Description worksheet for reference.
2. Decide if the new user needs to perform system functions. If so, add that information to the System Responsibilities form.
3. Add the person to the Individual User Profile form.
4. Review the System Responsibilities worksheet and the User Group Description worksheet to determine if the new user needs values that are different from those of the group.
5. Create a user profile by copying the group profile or the profile of a group member. Be sure to set the password to expire.
6. Give the new user a copy of your security memo.

Add a new application

This article describes how to add a new application, and provides step-by-step instructions.

You should plan the security for any new applications as carefully as you planned for your original applications. Follow the same procedures:

1. Prepare an Application Description worksheet and Library Description worksheet for the application.
2. Update your diagram of applications, libraries, and user groups.
3. Follow the procedures in "Planning resource security" to decide how to secure the new application.
4. Prepare an Application Installation worksheet by using the method described in "Planning your application installation."
5. Evaluate whether any printer output from the application is confidential and needs protection. Update your Output Queue and Workstation Security worksheet, if necessary.
6. Follow the steps described in "Setting up ownership and public authority" and "Setting up resource security" to install and secure the application.

Add a new workstation

This article describes how to add a new workstation, and provides step-by-step instructions.

When you add a new workstation to your system, consider security requirements:

1. Does the physical location of a new workstation pose any security risks? (See "Planning physical security" for more information.)
2. If the workstation does pose a risk, update your Output Queue and Workstation Security worksheet.
3. You should normally create new workstations with public authority *CHANGE. If this does not meet your security requirements for the workstation, use the EDTOBJAUT command to specify a different authority.

Change a user group

This article describes how to change a user group, explains why it is important, and provides step-by-step instructions.

You will need to handle different types of changes to the characteristics of a group in different ways. Following are some examples of changes and how to deal with them.

Changing the group's authority

You may discover that the group needs authority to objects that you did not anticipate in your initial planning. Do the following:

1. Use the Edit Object Authority (EDTOBJAUT) command to give the group the correct access to the objects or to an appropriate authorization list. "Setting up specific authorities" shows an example of how to do this. Every member of the group gets authority to the object when you give the group authority.
2. If you give the group authority to a confidential resource, you may want to verify the current members of the group. Use the Display User Profile command (DSPUSRPRF group-profile-name *GRPMBR) to list the group members.

Changing the customizing for the group

You may need to change the user environment setup for members of a group. For example, if a department gets its own printer, you want the new printer to be the default for the members of that department's user group. Or, when your system gets a new application installed, members of a user group may want a different initial menu when they sign on.

The group profile provides a pattern that you can copy to create individual profiles for group members. The customizing values in the group profile do not affect the individual user profiles after you create them, however. For example, changing a field, such as Printer device in the group profile, has no affect on the group members. You need to change the Printer device field in each individual user profile.

You can use the Work with User Profile display to change a parameter for more than one user at a time. The example shows changing the output queue for all members of a group:

1. Type WRKUSRPRF *ALL and press the Enter key.
2. If you see the Work with User Enrollment display, use F21 (Select assistance level) to change to the Work with User Profile display.

```
             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
     HARRISOK  Harrison, Keith
2    HOGANR    Hogan, Richard
     JONESS    Jones, Sharon
2    WILLISR   Willis, Rose
.
.
                                          More...
Parameters for options 1, 2, 3, 4 and 5 or command
===> PRTDEV(PRT02)
F3=Exit  F5=Refresh  F12=Cancel  F16=Repeat position to  F17=Position to
F21=Select assistance level  F24=More keys
```

3. Type a 2 (Change) next to each profile that you want to change.
4. On the parameter line at the bottom of the display, type the parameter name and the new value. If you do not know the parameter name, press F4 (Prompt).
5. Press the Enter key. You receive a confirmation message for each profile that changed. Although changing a customizing field in the group profile has no affect on the group members, it may help you in the future. The group profile provides a pattern when you want to add members to the group later. It is also a record of the standard field values for the group.

Giving the group access to a new application

When a user group needs access to a new application, you need to analyze information about the group and about the application. Following is a suggested method:

1. Look at the Application Description worksheet for the new application and your diagram of applications, libraries, and user groups to see which libraries the application uses. Add those libraries to the User Group Description worksheet.
2. Update your diagram of applications, libraries, and user groups to show the new relationship between the user group and application.
3. If the group's initial library list should include the libraries, change the group's job description by using the Change Job Description (CHGJOBD) command. See "Creating a job description" if you need help for working with job descriptions.

   Note: When you add libraries to the initial library list in a job description, you do not need to change the user profiles that use the job description. When the user signs on next, their initial library list automatically adds those libraries.
4. Evaluate whether you need to change either the initial program or the initial menu for the group to provide access to the new application. You need to make an individual change to the initial menu or program of each user profile by using the CHGUSRPRF command.
5. Review the Library Description forms for all the libraries that are used by the application. Determine whether the public access that is available for the libraries is sufficient for the group's needs. If it is not, you may need to give the group authority to the library, to specific objects, or to authorization lists. Use the Edit Object Authority (EDTOBJAUT) and the Edit Authorization List (EDTAUTL) commands to do this.

Change a user profile

This topic describes how to change a user profile, and provides step-by-step instructions.

When a system user gets a new job or a new set of responsibilities in your company, you need to evaluate how that affects the user profile.

1. Should the user belong to a different user group? You can use the CHGUSRPRF command to change the user profile.
2. Do you need to change any customizing values in the profile, such as the printer or the initial menu? You can use the CHGUSRPRF command to change these also.
3. Are the application authorities of the new user group sufficient for this person?
   - Use the Display User Profile (DSPUSRPRF) command to look at the authorities for the old and new group profiles.
   - Also look at the authorities of the individual user profile.
   - Make any changes necessary by using the EDTOBJAUT command.
4. Does the user own any objects? Should you change ownership of those objects? Use the Work with Objects by Owner (WRKOBJOWN) command.
5. Does the user perform system functions? Does the user need to perform system functions for the new job? Update the System Responsibilities worksheet and change the user profile, if necessary.

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.

Related concepts

"User profiles" on page 7
Every system user must have a user identity before they can sign on to and use a system. This user identity is called a user profile.

Enable a disabled user profile

This topic describes how to enable a disabled user profile, explains why it is important, and provides step-by-step instructions.

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

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

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

4. The main part of the ENABLEPGM program looks like this:

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

Listing User Profiles

You can display and print information about user profiles in a variety of formats. Displaying an Individual Profile To display the values for an individual user profile, use option **5** (Display) from either the Work with User Enrollment display or the Work with User Profiles display. Or, you can use the Display User Profile (DSPUSRPRF) command.

Listing All Profiles

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

```
             Display Authorized Users

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

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

```
             Display Authorized Users

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

Types of User Profile Displays

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

- Some displays and listings are available only for individual profiles. Others can be printed for all profiles or a generic set of profiles. Consult online information for details about the available types.

- You can create an output file from some displays by specifying *output*(*OUTFILE). Use a query tool or program to produce customized reports from the output file.

Related concepts

"User profiles" on page 7
Every system user must have a user identity before they can sign on to and use a system. This user identity is called a user profile.

Rename a user profile

This article describes how to rename a user profile, explains why it is important, and provides step-by-step instructions.

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 may not be able to change the information that is stored by applications that contain the user name.

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

The following example shows how to create a new profile for a user with a new name and the same authorities. The old profile name is SMITHM. The new user profile name is JONESM:

1. Copy the old profile (SMITHM) to a new profile (JONESM) using the copy option from the Work with User Enrollment display.
2. Give JONESM all the private authorities of SMITHM using the Grant User Authority (GRTUSRAUT) command:

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

   ```
   WRKOBJPGP PGP(SMITHM)
   ```

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

   Note: JONESM must have a gid assigned using the GID parameter on the Create or Change User Profile (CRTUSRPRF or CHGUSRPRF) command.
4. Display the SMITHM user profile using the Display User Profile (DSPUSRPRF) command: DSPUSRPRF USRPRF(SMITHM) Write down the uid and gid for SMITHM.
5. Transfer ownership of all other owned objects to JONESM and remove the SMITHM user profile, using option 4 (Remove) from the Work with User Enrollment display.
6. Change the uid and the gid of JONESM to the uid and gid that belonged to SMITHM by using the Change User Profile (CHGUSRPRF) command:

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

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

Schedule availability of user profiles

You may want some user profiles to be available for sign-on only at certain times of the day or certain days of the week.

For example, if you have a profile set up for a security auditor, you may want to enable that user profile only during the hours that the auditor is scheduled to work. You might also want to disable user profiles with *ALLOBJ special authority (including the QSECOFR user profile) during off-hours.

You can use the Change Activation Schedule Entry (CHGACTSCDE) command to set up user profiles to be enabled and disabled automatically. For each user profile that you want to schedule, you create an entry that defines the user profile's schedule.

For example, if you want the QSECOFR profile to be available only between 7 in the morning and 10 in the evening, you would type the following on the CHGACTSCDE display:

Figure 7. Schedule profile activation display—sample

```
         Change Activation Scd Entry (CHGACTSCDE)

Type choices, press Enter.

User profile . . . . . > QSECOFR         Name
Enable time  . . . . . > '7:00'          Time, *NONE
Disable time . . . . . > '22:00'         Time, *NONE
Days . . . . . > *MON            *ALL, *MON, *TUE, *WED...
                               > *TUE
                               > *WED
                               > *THU
              + for more values > *FRI
```

In fact, you might want to have the QSECOFR profile available only for a very limited number of hours each day. You can use another user profile with the *SECOFR class to perform most system functions. Thus, you avoid exposing a well-known user profile to hacking attempts.

You can use the Display Audit Journal Entries (DSPAUDJRNE) command periodically to print the CP (Change Profile) audit journal entries. Use these entries to verify that the system is enabling and disabling user profiles according to your planned schedule.

Another method for checking to ensure that user profiles are being disabled on your planned schedule is to use the Print User Profile (PRTUSRPRF) command. When you specify *PWDINFO for the report type, the report includes the status of each selected user profile. If, for example, you regularly disable all user profiles with *ALLOBJ special authority, you can schedule the following command to run immediately after the profiles are disabled: `PRTUSRPRF TYPE(*PWDINFO) SELECT(*SPCAUT) SPCAUT(*ALLOBJ)`

Remove a user from the system

This article describes how to remove a user from the system, explains why it is important, and provides step-by-step instructions.

If someone leaves your company, you should remove the user profile from the system immediately. Before you can delete a user profile, you must either delete or transfer ownership of any objects that are owned by the profile. You can use the WRKOBJOWN command to do this, or you can use option 4 (Remove) from the Work with User Enrollment display. When you select option 4 (Remove) for a profile from the Work with User Enrollment display, you see additional displays that allow you to handle any objects the user owns. You can choose to give all the objects to a new owner or handle the objects individually:

```
              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.
```

If you choose to handle the objects individually (option **2**), the screen displays a list of all the objects that are owned by the user:

```
              Remove User

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

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  Hogan, Richard message queue
 4   QUERY1  DPTWH    Inventory Query
```

If you choose to delete objects, you see the Confirm Delete display. Once the system deletes the objects, you can remove the user profile. You then see the Work with User Enrollment display again with a message that tells you that the system has removed the user.

Deleting User Profiles

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

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

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

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

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

Using the Delete User Profile Command

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

- All objects owned by the profile

- All objects for which the profile is the primary group
- EIM associations

Using the Remove User Option

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

```
              Remove User

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

To remove this user type a choice below, then press Enter.
  1. Give all objects owned by this user to a new owner
  2. Delete or change owner of specific objects owned by this user.
```

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

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

```
              Remove User

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

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

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

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

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

Note:

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.

Disable user profiles automatically:

If someone is gone from the organization for an extended period, disable (deactivate) that user's profile.

You can use the Analyze Profile Activity (ANZPRFACT) command to regularly disable user profiles that have been inactive for a specified number of days. When you use the ANZPRFACT command, you specify the number of inactive days that the system looks for. The system looks at the last used date, the restore date, and the creation date for the user profile.

Once you have specified a value for the ANZPRFACT command, the system schedules a job to run weekly at 1 a.m. (starting with the day after you first specified a value). The job examines all profiles and disables inactive profiles. You do not need to use the ANZPRFACT command again unless you want to change the number of inactive days.

You can use the Change Active Profile List (CHGACTPRFL) command to make some profiles exempt from ANZPRFACT processing. The CHGACTPRFL command creates a list of user profiles that the ANZPRFACT command will not disable, no matter how long those profiles have been inactive.

When the system runs the ANZPRFACT command, it writes a CP entry in the audit journal for each user profile that is disabled. You can use the DSPAUDJRNE command to list the user profiles that are newly disabled.

Remember: The system writes audit entries only if the QAUDCTL value specifies *AUDLVL and the QAUDLVL system value specifies *SECURITY.

Another method for checking to ensure that user profiles are being disabled on your planned schedule is to use the Print User Profile (PRTUSRPRF) command. When you specify *PWDINFO for the report type, the report includes the status of each selected user profile.

Remove user profiles automatically:

Your system should contain only user profiles that are necessary. An unnecessary user profile may provide unauthorized entry to your system. If you no longer need a user profile because the user either has left or has taken a different job within the organization, remove the user profile.

You can use the Change Expiration Schedule Entry (CHGEXPSCDE) command to manage the removing or disabling of user profiles. If you know that a user is leaving for an extended period, you can schedule the user profile to be removed or disabled.

The first time that you use the CHGEXPSCDE command, it creates a job schedule entry that runs at 1 minute after midnight every day. The job looks at the QASECEXP file to determine whether any user profiles are scheduled for removal on that day.

With the CHGEXPSCDE command, you either disable or delete a user profile. If you choose to delete a user profile, you must specify what the system will do with the objects that the user owns. Before you schedule a user profile for deletion, you need to research the objects that the user owns. For example, if the user owns programs that adopt authority, do you want those programs to adopt the ownership of the new owner? Or does the new owner have more authority than necessary (such as special authority)? Perhaps, you need to create a new user profile with specific authorities to own the programs that need to adopt authority.

You also need to research whether any application problems will occur if you delete the user profile. For example, do any job descriptions specify the user profile as the default user?

You can use the Display Expiration Schedule (DSPEXPSCD) command to display the list of profiles that are scheduled to be disabled or removed. You can use the Display Authorized Users (DSPAUTUSR) command to list all of the user profiles on your system. Use the Delete User Profile (DLTUSRPRF) command to delete outdated profiles.

Security note: You disable a user profile by setting its status to *DISABLED. When you disable a user profile, you make it unavailable for interactive use. You cannot sign on with or change your job to a disabled user profile. Batch jobs can run under a user profile that is disabled.

Configure the system to use security tools

This information describes how to set up your system to use the security tools that are part of i5/OS.

When you install i5/OS, the security tools are ready to use. The topics that follow provide suggestions for operating procedures with the security tools.

Use security tools securely

When you install i5/OS, the objects that are associated with the security tools are secure. To operate the security tools securely, avoid making authority changes to any security tool objects.

Following are the security settings and requirements for security tool objects:

- The security tool programs and commands are in the QSYS product library. The commands and the programs ship with the public authority of *EXCLUDE. Many of the security tool commands create files in the QUSRSYS library. When the system creates these files, the public authority for the files is *EXCLUDE. Files that contain information for producing changed reports have names that begin with QSEC. Files that contain information for managing user profiles have names that begin with QASEC. These files contain confidential information about your system. Therefore, you should not change the public authority to the files.
- The security tools use your normal system setup for directing printed output. These reports contain confidential information about your system. To direct the output to a protected output queue, make appropriate changes to the user profile or job description for users who will be running the security tools.
- Because of their security functions and because they access many objects on the system, the security tool commands require *ALLOBJ special authority. Some of the commands also require *SECADM, *AUDIT, or *IOSYSCFG special authority. To ensure that the commands run successfully, you should sign on as a security officer when you use the security tools. Therefore, you should not need to grant private authority to any security tool commands.

Avoid file conflicts

Many of the security tool report commands create a database file that you can use to print a changed version of the report. [Commands and menus for security commands] tells the file name for each command. You can only run a command from one job at a time. Most of the commands now have checks that enforce this. If you run a command when another job has not yet finished running it, you will receive an error message.

Many print jobs are long-running jobs. You need to be careful to avoid file conflicts when you submit reports to batch or add them to the job scheduler. For example, you might want to print two versions of the PRTUSRPRF report with different selection criteria. If you are submitting reports to batch, you should use a job queue that runs only one job at a time to ensure that the report jobs run sequentially.

If you are using the job scheduler, you need to schedule the two jobs far enough apart that the first version completes before the second job starts.

Related concepts

"System security tools" on page 17
You can use security tools to manage and monitor the security environment on your system.

Save security tools

You save the security tool programs whenever you run either the Save System (SAVSYS) command or an option from the Save menu that runs the SAVSYS command.

The security tool files are in the QUSRSYS library. You should already be saving this library as part of your normal operating procedures. The QUSRSYS library contains data for many licensed programs on your system. See the Information Center for more information about what commands and options save the QUSRSYS library.

Commands for customizing security

This section describes the commands and menus for security tools.

Commands and menus for security commands

Examples of how to use the commands are included throughout this information. 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.

Security Tools menu options

Table 112. Tool commands for user profiles

Menu option[1]	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]
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]

Table 112. Tool commands for user profiles (continued)

Menu option[1]	Command name	Description	Database file used
4	ANZPRFACT	Use the Analyze Profile Activity command to disable user profiles that have not been used for a specified number of days. After you use the ANZPRFACT command to specify the number of days, the system runs the ANZPRFACT job nightly. You can use the CHGACTPRFL command to exempt user profiles from being disabled.	QASECIDL[2]
5	DSPACTSCD	Use the Display Profile Activation Schedule command to display or print information about the schedule for enabling and disabling specific user profiles. You create the schedule with the CHGACTSCDE command.	QASECACT[2]
6	CHGACTSCDE	Use the Change Activation Schedule Entry command to make a user profile available for sign on only at certain times of the day or week. For each user profile that you schedule, the system creates job schedule entries for the enable and disable times.	QASECACT[2]
7	DSPEXPSCD	Use the Display Expiration Schedule command to display or print the list of user profiles that are scheduled to be disabled or removed from the system in the future. You use the CHGEXPSCDE command to set up user profiles to expire.	QASECEXP[2]

Table 112. Tool commands for user profiles (continued)

Menu option[1]	Command name	Description	Database file used
8	CHGEXPSCDE	Use the Change Expiration Schedule Entry command to schedule a user profile for removal. You can remove it temporarily (by disabling it) or you can delete it from the system. This command uses a job schedule entry that runs every day at 00:01 (1 minute after midnight). The job looks at the QASECEXP file to determine whether any user profiles are set up to expire on that day. Use the DSPEXPSCD command to display the user profiles that are scheduled to expire.	QASECEXP[2]
9	PRTPRFINT	Use the Print Profile Internals command to print a report containing information on the number of entries contained in a user profile. The number of entries determines the size of the user profile.	

Note:

1. Options are from the SECTOOLS menu.
2. This file is in the QUSRSYS library.

Values set by the Configure System Security command

This table lists the system values that are set when you run the CFGSYSSEC command. The CFGSYSSEC command runs a program that is called QSYS/QSECCFGS.

Values set by the CFGSYSSEC command

Table 113. Values set by the CFGSYSSEC command

System value names	Setting	System value description
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

Table 113. Values set by the CFGSYSSEC command (continued)

System value names	Setting	System value description
QDSPSGNINF	1 (Yes)	Whether users see the sign-on information display
QINACTITV	60	Time period before the system takes action on an inactive interactive job
QINACTMSGQ	*ENDJOB	Action that the system takes for an inactive job
QLMTDEVSSN	1 (Yes)	Whether users are limited to signing on at one device at a time
QLMTSECOFR	1 (Yes)	Whether *ALLOBJ and *SERVICE users are limited to specific devices
QMAXSIGN	3	How many consecutive, unsuccessful sign-on attempts are allowed
QMAXSGNACN	3 (Both)	Whether the system disables the workstation or the user profile when the QMAXSIGN limit is reached.
QRMTSIGN	*FRCSIGNON	How the system handles a remote (pass-through or TELNET) sign-on attempt.
QRMTSVRATR	0 (Off)	Allows the system to be analyzed remotely.
QSECURITY	50	The level of security that is enforced
QVFYOBJRST	3 (Verify signatures on restore)	Verify object on restore
QPWDEXPITV	60	How often users must change their passwords
QPWDMINLEN	6	Minimum length for passwords
QPWDMAXLEN	8	Maximum length for passwords
QPWDPOSDIF	1 (Yes)	Whether every position in a new password must differ from the same position in the last password
QPWDLMTCHR		Characters that are not allowed in passwords
QPWDLMTAJC	1 (Yes)	Whether adjacent numbers are prohibited in passwords
QPWDLMTREP	2 (Cannot be repeated consecutively)	Whether repeating characters in are prohibited in passwords
QPWDRQDDGT	1 (Yes)	Whether passwords must have at least one number
QPWDRQDDIF	1 (32 unique passwords)	How many unique passwords are required before a password can be repeated
QPWDVLDPGM	*NONE	The user exit program that the system calls to validate passwords

Note:

1. The restricted characters are stored in message ID CPXB302 in the message file QSYS/QCPFMSG. They are shipped as AEIOU@$#. You can use the Change Message Description (CHGMSGD) command to change the restricted characters. The QPWDLMTCHR system value is not enforced at password levels 2 or 3.

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 using the Change Security Auditing (CHGSECAUD) command. The CFGSYSSEC command turns on action and object auditing and also, specifies the default set of actions to audit on the CHGSECAUD command.

Customize the program:

If some of these settings are not appropriate for your installation, you can create your own version of the program that processes the command.

Do the following:

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 would type the following: `CHGCMD CMD(QSYS/CFGSYSSEC) PGM(QGPL/MYSECCFG)`

   Note: If you change the QSYS/QSECCFGS program, IBM cannot guarantee or imply reliability, serviceability, performance or function of the program. The implied warranties of merchantability and fitness for a particular purpose are expressly disclaimed.

Functions of the Revoke Public Authority command

You can use the Revoke Public Authority (RVKPUBAUT) command to set the public authority to *EXCLUDE for a set of commands and programs.

Commands and APIs whose public authority are set by the RVKPUBAUT command

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 the table below and the application programming interfaces (APIs) that are listed in Table 12. When your system arrives, these commands and APIs have their public authority set to *USE.

The commands and the APIs that are listed in the tables all perform functions on your system that may provide an opportunity for mischief. As security administrator, you should explicitly authorize users to run these commands and programs rather than make them available to all system users.

When you run the RVKPUBAUT command, you specify the library that contains the commands. The default is the QSYS library. If you have more than one national language on your system, you need to run the command for each QSYSxxx library.

Table 114. Setting public authority

Using the RVKPUBAUT command		
ADDAJE	CHGJOBQE	RMVCMNE
ADDCFGLE	CHGPJE	RMVJOBQE
ADDCMNE	CHGRTGE	RMVPJE
ADDJOBQE	CHGSBSD	RMVRTGE
ADDPJE	CHGWSE	RMVWSE
ADDRTGE	CPYCFGL	RSTLIB
ADDWSE	CRTCFGL	RSTOBJ
CHGAJE	CRTCTLAPPC	RSTS36F
CHGCFGL	CRTDEVAPPC	RSTS36FLR
CHGCFGLE	CRTSBSD	RSTS36LIBM
CHGCMNE	ENDRMTSPT	STRRMTSPT
CHGCTLAPPC	RMVAJE	STRSBS
CHGDEVAPPC	RMVCFGLE	WRKCFGL

Table 115. Setting public authority

Using the RVKPUBAUT command
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).

Customize the program:

If some of these settings are not appropriate for your installation, you can create your own version of the program that processes the command.

Do the following:

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 would type the following: `CHGCMD CMD(QSYS/RVKPUBAUT) PGM(QGPL/MYRVKPGM)`

   Note: If you change the QSYS/QSECRVKP program, IBM cannot guarantee or imply reliability, serviceability, performance or function of the program. The implied warranties of merchantability and fitness for a particular purpose are expressly disclaimed.

Use security exit programs

Some system functions provide an exit so that your system can run a user-created program to perform additional checking and validation. For example, you can set up your system to run an exit program every time that someone attempts to open a DDM (distributed data management) file on your system.

Sources of Sample Exit Programs

You can use the registration function to specify exit programs that run under certain conditions. The following table provides a list of these exit programs and sources for example programs.

Table 116. Sources of Sample Exit Programs

Type of exit programs	Purpose	Where to find examples
Password validation	The QPWDVLDPGM system value can specify a program name or indicate that validation programs registered for the QIBM_QSY_VLD_PASSWRD exit point be used to check a new password for additional requirements that are not handled by the QPWDxxx system values. The use of this program should be carefully monitored because it receives unencrypted passwords. This program should not store passwords in a file or pass them to another program.	• An Implementation Guide for iSeries Security and Auditing, GG24–4200 • iSeries Security Reference, SC41-5302-07
PC Support/400 or Client Access access[1]	You can specify this program name in the Client request access (PCSACC) parameter of the network attributes to control the following functions: • Virtual printer function • File transfer function v Shared folders Type 2 function • Client access message function • Data queues • Remote SQL function	An Implementation Guide for iSeries Security and Auditing, GG24–4200
Distributed Data Management (DDM) access	You can specify this program name in the DDM request access (DDMACC) parameter of the network attributes to control the following functions: • Shared folders Type 0 and 1 function • Submit Remote Command function	An Implementation Guide for iSeries Security and Auditing, GG24–4200
Remote sign on	You can specify a program in the QRMTSIGN system value to control what users can be automatically signed on from which locations (pass-through.)	An Implementation Guide for iSeries Security and Auditing, GG24–4200

Table 116. Sources of Sample Exit Programs (continued)

Type of exit programs	Purpose	Where to find examples
Open Database Connectivity (ODBC) with iSeries Access[1]	Control the following functions of ODBC: • Whether ODBC is allowed at all. • What functions are allowed for iSeries database files. • What SQL statements are allowed. • What information can be retrieved about database server objects. • What SQL catalog functions are allowed.	None available
QSYSMSG break handling program	You can create a program to monitor the QSYSMSG message queue and take appropriate action (such as notifying the security administrator) depending on the type of message.	An Implementation Guide for iSeries Security and Auditing, GG24–4200
TCP/IP	Several TCP/IP servers (such as FTP, TFTP, TELNET, and REXEC) provide exit points. You can add exit programs to handle log-on and to validate user requests, such as requests to get or put a specific file. You can also use these exits to provide anonymous FTP on your system.	TCP/IP User Exits in the iSeries System API Reference book
User profile changes	You can create exit programs for the following user profile commands: CHGUSRPRF CRTUSRPRF DLTUSRPRF RSTUSRPRF	• iSeries Security Reference, SC41-5302-07 • TCP/IP User Exits in the iSeries System API Reference book

Manage service tools user ID

This article describes how to manage service tool user IDs using DST, SST, and iSeries Navigator.

Service tools are used to configure, manage, and service your server. Service tools can be accessed from dedicated service tools (DST) or system service tools (SST). Service tools user IDs are required to access DST, SST, and to use iSeries Navigator functions for logical partition (LPAR) management and disk unit management. DST is available when the Licensed Internal Code has been started, even if i5/OS has not been loaded. SST is available from i5/OS. The following table outlines the basic differences between DST and SST.

Characteristic	DST	SST
How to access	Physical access through console during a manual IPL or by selecting option 21 on the control panel.	Access through interactive job with the ability to sign on with QSRV or the following authorizations: • Authorized to STRSST (Start SST) CL command. • Service special authority (*SERVICE) or all object special authority (*ALLOBJ). • Functional privilege to use SST.

Characteristic	DST	SST
When available	Available even when the server has limited capabilities. i5/OS is not required to access DST.	Available when i5/OS has been started. i5/OS is required to access SST.
How to authenticate	Requires service tools user ID and password.	Requires service tools user ID and password.

See the **iSeries Information Center → Security → Service** tools for information about using the Service tools to perform the following tasks:

- Access service tools with DST
- Access service tools with SST
- Access service tools with iSeries Navigator
- Create a service tools user ID using DST
- Change the functional privileges for a service tools user ID using DST
- Change the description for a service tools user ID using DST
- Display a service tools user ID using DST
- Enable a service tools user using DST
- Disable a service tools user using DST
- Delete a service tools user ID using DST
- Change service tools user IDs and passwords using SST or DST
- Change your service tools user IDs and password using STRSST or QSYCHGDS API
- Reset or recover the QSECOFR user profile password
- Reset the QSECOFR service tools user ID and password
- Save and restore service tools security data
- Configure the service tools server for DST
- Configure the service tools server for i5/OS
- Monitor service function use through DST
- Monitor service tools use through i5/OS security audit log

Protect against computer viruses

This article provides some tips for protecting against computer viruses and suspicious programs.

Recent trends in computer usage have increased the likelihood that your system has programs from untrusted sources or programs that perform unknown functions. Following are examples:

- A personal computer user sometimes obtains programs from other PC users. If the PC is attached to your system, that program can affect your server.
- Users who connect to networks can also obtain programs, for example from bulletin boards.
- Hackers have become more active and renowned. They often publish their methods and their results. This can lead to imitation by normally law-abiding programmers.

These trends have led to a problem in computer security that is called a **computer virus**. A virus is a program that can change other programs to include a copy of itself. The other programs are then said to be infected by the virus. Additionally, the virus can perform other operations that can take up system resources or destroy data.

The architecture of the server provides some protection from the infectious characteristics of a computer virus. “Protect against computer viruses” describes this. A server security administrator needs to be more concerned about programs that perform unauthorized functions. The remaining topics in this chapter

describe ways that someone with ill intentions might set up harmful programs to run on your system. The topics provide tips for preventing programs from performing unauthorized functions.

Tip: Object authority is always your first line of defense. If you do not have a good plan for protecting your objects, your system is defenseless. This information discusses ways that an authorized user might try to take advantage of loopholes in your object authority scheme.

A computer that has a virus infection has a program that can change other programs. The object-based architecture of this system makes it more difficult for a mischief-maker to produce and spread this type of virus than it is with other computer architectures. On this system, you use specific commands and instructions to work on each type of object. You cannot use a file instruction to change an operable program object (which is what most virus-creators do). Nor can you easily create a program that changes another program object. To do this requires considerable time, effort, and expertise, and it requires access to tools and documentation that are not generally available.

However, as new server functions become available to participate in the open-systems environment, some of the object-based protection functions of servers no longer apply. For example, with the integrated file system (IFS), users can directly manipulate some objects in directories, such as stream files.

Also, although server architecture makes it difficult for a virus to spread among server programs, its architecture does not prevent the system from being a virus-carrier. As a file server, the server can store programs that many PC users share. Any one of these programs might contain a virus that the server does not detect. To prevent this type of virus from infecting the PCs that are attached to your server, you must use PC virus-scan software. Several functions exist on the server to prevent someone from using a low-level language with pointer capability to alter an operable object program:

- If your system runs at security level 40 or higher, the integrity protection includes protections against changing program objects. For example, you cannot successfully run a program that contains blocked (protected) machine instructions.
- The program validation value is also intended to protect you when you restore a program that was saved (and potentially changed) on another system. Chapter 2 in the *iSeries Security Reference* describes the integrity protection functions for security level 40 and higher, including program validation values.

Note: The program validation value is not foolproof, and it is not a replacement for vigilance in evaluating programs that are restored to your system.

Several tools are also available to help you detect the introduction of an altered program into your system:

- You can use the Check Object Integrity (CHKOBJITG) command to scan objects (operable objects) that meet your search values to ensure that those objects have not been altered. This is similar to a virus-scan function. You can also use the CHKOBJITG command to request that a scan be done of the integrated file system objects. If a user has an application or business partner that scans for viruses using the integrated file system's scan related exit programs, this will trigger a scan for viruses.
- You can use the security auditing function to monitor programs that are changed or restored. The *PGMFAIL, *SAVRST, and *SECURITY values for the authority level system value provide audit records that can help you detect attempts to introduce a virus-type program into your system. Chapter 9 and Appendix F in the *Security Reference* provide more information about audit values and the audit journal entries.
- You can use the force create (FRCCRT) parameter of the Change Program (CHGPGM) command to recreate any program that has been restored to your system. The system uses the program template to recreate the program. If the program object has been changed after it was compiled, the system recreates the changed object and replaces it. If the program template contains blocked (protected) instructions, the system will not recreate the program successfully.

- You can use the QFRCCVNRST (force conversion on restore) system value to recreate any program as it restored to your system. The system uses the program template to recreate the program. This system value provides several choices on which programs to recreate.
- You can use the QVFYOBJRST (verify objects on restore) system value to prevent the restore of programs that do not have a digital signature or do not have a valid digital signature. When a digital signature is not valid, it means the program has been changed since it was signed by its developer. APIs exist that allow you to sign your own programs, save files, and stream files

Use the Print Publicly Authorized Objects command (PRTPUBAUT)

This article describes how to use the print publicly authorized objects commands (PRTPUBAUT), explains why it is important, and provides step-by-step instructions.

The Print Publicly Authorized Objects (PRTPUBAUT) command allows you to print a report of the specified objects that do not have public authority of *EXCLUDE. For *PGM objects, only the programs that do not have public authority of *EXCLUDE that a user can call (the program is either user domain or the system security level (QSECURITY system value) is 30 or below) will be included in the report. This is a way to check for objects that every user on the system is authorized to access.

This command will print two reports. The first report (Full Report) will contain all of the specified objects that do not have public authority of *EXCLUDE. The second report (Changed Report) will contain the objects that now do not have public authority of *EXCLUDE that did have public authority of *EXCLUDE or did not exist when the PRTPUBAUT command was previously run. If the PRTPUBAUT command was not previously run for the specified objects and library, folder, or directory, there will be no 'Changed Report'. If the command has been previously run, but no additional objects do not have public authority of *EXCLUDE, then the 'Changed Report' will be printed but there will be no objects listed.

Restriction: You must have *ALLOBJ special authority to use this command.

Examples: This command creates the full, and changed reports for all the file objects in the library GARRY that do not have a public authority of *EXCLUDE:

```
PRTPUBAUT OBJTYPE(*FILE) LIB(GARRY)
```

This command creates the full, changed, and deleted reports for all the stream file objects in the subdirectory structure that starts at the directory garry that do not have a public authority of *EXCLUDE:

```
PRTPUBAUT OBJTYPE(*STMF) DIR(GARRY) SCHSUBDIR(*YES)
```

Use the Print Private Authorities command (PRTPVTAUT)

This article describes how to use the print private authorities command (PRTPVTAUT), explains why it is important, and provides step-by-step instructions.

The Print Private Authorities (PRTPVTAUT) command allows you to print a report of all the private authorities for objects of a specified type in a specified library, folder, or directory. The report lists all objects of the specified type and the users that are authorized to the object. This is a way to check for different sources of authority to objects.

This command prints three reports for the selected objects. The first report (Full Report) contains all of the private authorities for each of the selected objects. The second report (Changed Report) contains additions and changes to the private authorities to the selected objects if the PRTPVTAUT command was previously run for the specified objects in the specified library, folder, or directory. Any new objects of the selected type, new authorities to existing objects, or changes to existing authorities to the existing objects are listed in the 'Changed Report'. If the PRTPVTAUT command was not previously run for the specified

objects in the specified library, folder, or directory, there will be no 'Changed Report'. If the command has been previously run but no changes have been made to the authorities on the objects, then the 'Changed Report' is printed but there are no objects listed.

The third report (Deleted Report) contains any deletions of privately authorized users from the specified objects since the PRTPVTAUT command was previously run. Any objects that were deleted or any users that were removed as privately authorized users are listed in the 'Deleted Report'. If the PRTPVTAUT command was not previously run, there will be no 'Deleted Report'. If the command has been previously run but no delete operations have been done to the objects, then the 'Deleted Report' is printed but there are no objects listed.

Restriction: You must have *ALLOBJ special authority to use this command.

Examples: This command creates the full, changed, and deleted reports for all file objects in the PAYROLLLIB:

```
PRTPVTAUT OBJTYPE(*FILE) LIB(PAYROLLLIB)
```

This command creates the full, changed, and deleted reports for all the stream file objects in the directory garry:

```
PRTPVTAUT OBJTYPE(*STMF) DIR(/GARRY) SCHSUBDIR(*NO)
```

This command creates the full, changed, and deleted reports for all the stream file objects in the subdirectory structure that starts at the directory garry:

```
PRTPVTAUT OBJTYPE(*STMF) DIR(/GARRY) SCHSUBDIR(*YES)
```

Print user profile reports

The following commands provide user profile reports.

- Print User Profile (PRTUSRPRF)

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

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

Use the Print System Security Attributes command (PRTSYSSECA)

Purpose

This example shows the output from the Print System Security Attributes (PRTSYSSECA) command. The report shows the settings for security-relevant system values and network attributes that are recommended for systems with normal security requirements. It also shows the current settings on your system.

Note: The Current Value column on the report shows the current setting on your system. Compare this value to the recommended value to see where you may have security exposures.

Sample System Security Attributes Report

```
System Security Attributes

System Value
```

```
Name    Current Value  Recommended Value
QALWOBJRST *NONE      *NONE
QALWUSRDMN *ALL      QTEMP
QATNPGM   QEZMAIN QSYS   *NONE
QAUDENDACN *NOTIFY     *NOTIFY
QAUDFRCLVL *SYS      *SYS
QAUDCTL   *AUDLVL     *AUDLVL  *OBJAUD
QAUDLVL   *SECURITY    *AUTFAIL *CREATE
            *DELETE  *SECURITY
            *SAVRST  *NOQTEMP

QAUTOCFG  0       0
QAUTORMT  1       0
QAUTOVRT  9999      0
QCMNRCYLMT 0 0      0 0
QCRTAUT   *CHANGE     Control at library level
QCRTOBJAUD *NONE     Control at library level
QDEVRCYACN *DSCMSG     *DSCMSG
QDSCJOBITV 120      120
QDSPSGNINF 1       1
QINACTITV  60       60
QINACTMSGQ *ENDJOB     *ENDJOB
QLMTDEVSSN 0       1
QLMTSECOFR 0       1
QMAXSGNACN 2       3
QMAXSIGN  3       3

QPWDEXPITV 60       60
QPWDLMTAJC 1       1
QPWDLMTCHR *NONE     AEIOU@$#
QPWDLMTREP 1       2
QPWDLVL   0
QPWDMAXLEN 8       8
QPWDMINLEN 6       6
QPWDPOSDIF 1       1
QPWDRQDDGT 1       1
QPWDRQDDIF 0       1
QPWDVLDPGM *NONE     *NONE
QRETSVRSEC 0       0
QRMTIPL   0       0
QRMTSIGN  *FRCSIGNON   *FRCSIGNON
QSECURITY  50       50
QSHRMEMCTL 1       0
QSRVDMP   *DMPUSRJOB   *NONE
QUSEADPAUT *NONE     CRTAUTL AUTL(QUSEADPAUT) AUT(*EXCLUDE)
            CHGOBJOWN OBJ(QUSEADPAUT) OBJTYPE(*AUTL)
            CHGSYSVAL SYSVAL(QUSEADPAUT) VALUE(QUSEADPAUT)
QVFYOBJRST 1       3
Network Attribute
Name   Current Value   Recommended Value
DDMACC  *OBJAUT      *REJECT
JOBACN  *FILE      *REJECT
PCSACC  *OBJAUT      *REJECT
```

Monitor security

This set of topics discuss various techniques for monitoring and auditing security on your system.

In a **security audit**, you would be reviewing and examining the activities of a data processing system to test the adequacy and effectiveness of procedures for data security and data accuracy. The **security audit journal** is the primary source of auditing information on the system. A security auditor inside or outside your organization can use the auditing function provided by the system to gather information about security-related events that occur on the system.

An **intrusion detection** system is software that detects attempts or successful attacks on monitored resources that are part of a network or host system.

Monitoring security regularly has two basic goals:
- Making sure that you protect your company resources adequately.
- Detecting unauthorized attempts to access your system and your company's information.

This set of topics describes the tasks for auditing and monitoring system security:

Plan security auditing

Use this information to plan security auditing for your systems.

When monitoring your security, the operating system can log security events which occur on your system. These events are recorded in special system objects called journal receivers. You can set up journal receivers to record different types of security events, such as changing a system value or user profile, or an unsuccessful attempt to access an object. The following values control which events are logged:
- The audit control (QAUDCTL) system value
- The audit level (QAUDLVL) system value
- The audit level (AUDLVL) value in user profiles
- The object auditing (OBJAUD) value in user profiles and objects

The information in the audit journals is used:
- To detect attempted security violations.
- To plan migration to a higher security level.
- To monitor the use of sensitive objects, such as confidential files.

Commands are available to view the information in the audit journals in different ways.

The purpose of an audit is to detect and log activities that might compromise the security of your system. When you choose to log actions that occur on your systems, you might experience a trade-off in performance and, in some cases, loss of disk space. If you decide to log security-related events on your systems, the eServer Security Planner will provide some recommendations about what level of auditing you should do.

To plan the use of security auditing on your system, follow these steps:
- Use the eServer Security Planner to see what it recommends about what level of auditing you should do based on your system configuration and user requirements.
- 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.

The security audit journal is the primary source of auditing information on the system. A security auditor inside or outside your organization can use the auditing function provided by the system to gather information about security-related events that occur on the system. You use system values, user profile parameters, and object parameters to define auditing.

The security auditing function is optional. You must take specific steps to set up security auditing.

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.

When a security-related event that may be audited occurs, the system checks whether you have selected that event for audit. If you have, the system writes a journal entry in the current receiver for the security auditing journal (QAUDJRN in library QSYS).

For information on planning the auditing of actions and auditing of object access, see Chapter 9 of the iSeries Security Reference.

Related concepts

"Security audits" on page 17
This topic describes the purpose of security audits.

Checklists for security auditing

Use this checklist to plan and audit system security.

As you plan security, choose the items from the list that meet your security requirements. When you audit the security of your system, use the list to evaluate the controls you have in place and to determine if additional controls are needed. The list contains brief descriptions of how to do each item and how to monitor that it has been done.

Table 117. Security Auditing Planning Form

Security Auditing Planning Form	
Prepared by:	Date:
Monitoring physical security:	
Is backup media protected from damage and theft?	
Is access to workstations in public areas restricted? Use the DSPOBJAUT command to see who has *CHANGE authority to the workstations.	
Monitoring system values:	
Verify that the settings for system values match your System Values Selection form. Use the Print System Security Attributes (PRTSYSSECA) command.	
Review your decisions about system values, particularly when you install new applications. Have any system values changed?	
Monitoring group profiles:	
Verify that group profiles have no passwords. Use the DSPAUTUSR command to verify that all group profiles have a password of *NONE.	
Verify that the correct people are members of the group. Use the DSPUSRPRF command with the *GRPMBR option to list the members of a group.	
Check the special authorities for each group profile. Use the DSPUSRPRF command. If you are running at security level 30, 40, or 50, group profiles should not have *ALLOBJ authority.	
Monitoring user profiles:	

Table 117. Security Auditing Planning Form (continued)

Security Auditing Planning Form	
Verify that user profiles on the system belong to one of these categories: • User profiles for current employees • Group profiles • Application owner profiles • IBM-supplied profiles (start with Q)	
Remove their user profile when the company transfers a user or when a user leaves the company. Use the Change Expiration Schedule Entry (CHGEXPSCDE) command to automatically delete or disable the profile as soon as the user leaves.	
Look for inactive profiles and remove them. Use the Analyze Profile Activity (ANZPRFACT) command to automatically disable profiles after they have been inactive for a certain time.	
Determine which users have a password that is the same as their user profile name. Use the Analyze Default Passwords (ANZDFTPWD) command. Use the option of this command to force users to change their passwords the next time they sign on to the system. **Attention:** Do not remove any IBM-supplied profiles from the system. IBM-supplied profiles start with the character Q.	
Be aware of who has a user class other than *USER and why. Use the Print User Profile (PRTUSRPRF) command to get a list of all users, their user class, and their special authorities. Match this information with your System Responsibilities form.	
Control which user profiles have the Limit capabilities field set to *NO.	
Monitoring critical objects:	
Review who has access to critical objects. Use the Print Private Authorities (PRTPVTAUT) command and the Print Publicly Authorized Objects (PRTPUBAUT) command to monitor objects. If a group has access, verify the members of the group with the *GRPMBR option of the DSPUSRPRF command.	
Verify who can use application programs that provide access to objects through another security method, such as adopted authority. Use the Print Adopting Objects (PRTADPOBJ) command.	
Monitoring unauthorized access:	
Instruct system operators to be alert for security messages in the QSYSOPR message queue. In particular, have them notify a security officer of repeated unsuccessful attempts to sign on. Security messages are in the range of 2200 to 22FF and 4A00 to 4AFF. They have prefixes CPF, CPI, CPC, and CPD.	
Set up security auditing to log unauthorized attempts to access objects.	

For additional information on using the security auditing checklist, see Chapter 9 of the iSeries Security Reference.

Set up security auditing

This article describes how to set up security auditing, explains why it is important, and provides step-by-step instructions. The system collects security events in the QAUDJRN journal.

Setting up auditing requires *AUDIT special authority. To set up security auditing, do the following 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) +
             TEXT('Auditing Journal Receiver')
   ```

   - Place the journal receiver in a library that is saved regularly. Do not place the journal receiver in library QSYS, even though that is where the journal will be.
   - Choose a journal receiver name that can be used to create a naming convention for future journal receivers, such as AUDRCV0001. You can use the *GEN option when you change journal receivers to continue the naming convention. Using this type of naming convention is also useful if you choose to have the system manage changing your journal receivers.
   - Specify a receiver threshold appropriate to your system size and activity. The size you choose should be based on the number of transactions on your system and the number of actions you choose to audit. If you use system change-journal management support, the journal receiver threshold must be at least 100 000 KB.
   - Specify *EXCLUDE on the AUT parameter to limit access to the information stored in the journal.
2. Create the QSYS/QAUDJRN journal by using the Create Journal (CRTJRN) command:

   ```
   CRTJRN JRN(QSYS/QAUDJRN) +
          JRNRCV(JRNLIB/AUDRCV0001) +
          MNGRCV(*SYSTEM) DLTRCV(*NO) +
          AUT(*EXCLUDE) TEXT('Auditing Journal')
   ```

   - The name QSYS/QAUDJRN *must* be used.
   - Specify the name of the journal receiver you created in the previous step.
   - Specify *EXCLUDE on the AUT parameter to limit access to the information stored in the journal. You must have authority to add objects to QSYS to create the journal.
   - Use the Manage receiver (MNGRCV) parameter to have the system change the journal receiver and attach a new one when the attached receiver exceeds the threshold specified when the journal receiver was created. If you choose this option, you do not have to use the CHGJRN command to detach receivers and create and attach new receivers manually.
   - Do not have the system delete detached receivers. Specify DLTRCV(*NO), which is the default. The QAUDJRN receivers are your security audit trail. Ensure that they are adequately saved before deleting them from the system.
3. Set the audit level (QAUDLVL) system value or the audit level extension (QAUDLVL2) system value 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.
4. Set action auditing for individual users if necessary using the CHGUSRAUD command.
5. Set object auditing for specific objects if necessary using the CHGOBJAUD and CHGDLOAUD commands.
6. Set object auditing for specific users if necessary using the CHGUSRAUD command.
7. Set the QAUDENDACN system value to control what happens if the system cannot access the audit journal.
8. Set the QAUDFRCLVL system value to control how often audit records are written to auxiliary storage.
9. Start auditing by setting the QAUDCTL system value to a value other than *NONE.

Note: The QSYS/QAUDJRN journal must exist before you can change the QAUDCTL system value to a value other than *NONE. When you start auditing, the system attempts to write a record to the audit journal. If the attempt is not successful, you receive a message and auditing does not start.

For more information, see the following topics in the iSeries Security Reference:
"Planning the Auditing of Actions"
"Planning the Auditing of Object Access"
"Audit End Action"

Use the security audit journal

The security audit journal is the primary source of auditing information on the system. A security auditor inside or outside your organization can use the auditing function provided by the system to gather information about security-related events that occur on the system.

The information in the **audit journals** is used:

- To detect attempted security violations.
- To plan migration to a higher security level.
- To monitor the use of sensitive objects, such as confidential files.

Commands are available to view the information in the audit journals in different ways. 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.

When monitoring your security, the operating system can log security events which occur on your system. These events are recorded in special system objects called **journal receivers**. You can set up journal receivers to record different types of security events, such as changing a system value or user profile, or an unsuccessful attempt to access an object. The following values control which events are logged:

- The audit control (QAUDCTL) system value
- The audit level (QAUDLVL) system value
- The audit level (AUDLVL) value in user profiles
- The object auditing (OBJAUD) value in user profiles
- The object auditing (OBJAUD) value in objects

Manage the audit journal and journal receivers

The auditing journal, QSYS/QAUDJRN, is intended solely for security auditing. Objects should not be journaled to the audit journal. Commitment control should not use the audit journal. User entries should not be sent to this journal using the Send Journal Entry (SNDJRNE) command or the Send Journal Entry (QJOSJRNE) API.

Special locking protection is used to ensure that the system can write audit entries to the audit journal. When auditing is active (the QAUDCTL system value is not *NONE), the system arbitrator job (QSYSARB) holds a lock on the QSYS/QAUDJRN journal. You cannot perform certain operations on the audit journal when auditing is active, such as:

- DLTJRN command
- ENDJRN*xxx* command
- APYJRNCHG command
- RMVJRNCHG command
- DMPOBJ or DMPSYSOBJ command
- Moving the journal
- Restoring the journal
- Operations that work with authority, such as the GRTOBJAUT command

- WRKJRN command

The information recorded in the security journal entries is described in Security Reference book. All security entries in the audit journal have a journal code of T. In addition to security entries, system entries also appear in the journal QAUDJRN. These are entries with a journal code of J, which relate to initial program load (IPL) and general operations performed on journal receivers (for example, saving the receiver).

If damage occurs to the journal or to its current receiver so that the auditing entries cannot be journaled, the QAUDENDACN system value determines what action the system takes. Recovery from a damaged journal or journal receiver is the same as for other journals.

You may want to have the system manage the changing of journal receivers. Specify MNGRCV(*SYSTEM) when you create the QAUDJRN journal, or change the journal to that value. If you specify MNGRCV(*SYSTEM), the system automatically detaches the receiver when it reaches its threshold size and creates and attaches a new journal receiver. This is called **system change-journal management**.

Analyze object authorities

This article describes how to analyze object authorities and provides step-by-step instructions.

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(QSYS/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(QSYS/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.

Analyze programs that adopt authority

This article describes the step-by-step procedure for analyzing programs that adopt authority.

Programs that adopt the authority of a user with *ALLOBJ special authority represent a security exposure. The following method can be used to find and inspect those programs:

1. For each user with *ALLOBJ special authority, use the Display Programs That Adopt (DSPPGMADP) command to list the programs that adopt that user's authority:
   ```
   DSPPGMADP USRPRF(user-profile-name) +
             OUTPUT(*PRINT)
   ```
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(library-name/program-name) +
             OUTPUT(*PRINT)
   ```
3. Inspect the source code and program description to evaluate:
   - Whether the user of the program is prevented from excess function, such as using a command line, while running under the adopted profile.
   - Whether the program adopts the minimum authority level needed for the intended function. Applications that use program failure can be designed using the same owner profile for objects and programs. When the authority of the program owner is adopted, the user has *ALL authority to application objects. In many cases, the owner profile does not need any special authorities.
4. Verify when the program was last changed, using the DSPOBJD command:

```
DSPOBJD OBJ(library-name/program-name) +
        OBJTYPE(*PGM) ASPDEV(library-name/program-name) +
        DETAIL(*FULL)
```

Analyze user profiles

This article describes how to analyze user profiles and provides step-by-step instructions.

You can display or print a complete list of all the users on your system with the Display Authorized Users (DSPAUTUSR) command. The list can be sequenced by profile name or group profile name. Following is an example of the group profile sequence:

```
             Display Authorized Users
                  Password
Group     User      Last      No
Profile   Profile   Changed   Password  Text

DPTSM
          ANDERSOR  08/04/0x            Roger Anders
          VINCENTM  09/15/0x            Mark Vincent
DPTWH
          ANDERSOR  08/04/0x            Roger Anders
          WAGNERR   09/06/0x            Rose Wagner
QSECOFR
          JONESS    09/20/0x            Sharon Jones
          HARRISOK  08/29/0x            Ken Harrison
*NO GROUP
          DPTSM     09/05/0x    X       Sales and Marketing
          DPTWH     08/13/0x    X       Warehouse
          RICHARDS  09/05/0x            Janet Richards
          SMITHJ    09/18/0x            John Smith
```

Print selected user profiles

You can use the Display User Profile (DSPUSRPRF) command to create an output file, which you can process using a query tool.

```
DSPUSRPRF USRPRF(*ALL) +
          TYPE(*BASIC) OUTPUT(*OUTFILE)
```

You can use a query tool to create a variety of analysis reports of your output file, such as:

- A list of all users who have both *ALLOBJ and *SPLCTL special authority.
- A list of all users sequenced by a user profile field, such as initial program or user class.

You can create query programs to produce different reports from your output file. For example:

- List all user profiles that have any special authorities by selecting records where the field UPSPAU is not equal to *NONE.
- List all users who are allowed to enter commands by selecting records where the Limit capabilities field (called UPLTCP in the model database outfile) is equal to *NO or *PARTIAL.
- List all users who have a particular initial menu or initial program.
- List inactive users by looking at the date last sign-on field.
- List all users who do not have a password for use at password levels 0 and 1 by selecting records where the Password present for level 0 or 1 field (called UPENPW in the model outfile) is equal to N.
- List all users who have a password for use at password levels 2 and 3 by selecting records where the Password present for level 2 or 3 field (called UPENPH in the model outfile) is equal to Y.

Examine large user profiles

User profiles with large numbers of authorities, appearing to be randomly spread over most of the system, can reflect a lack of security planning. Following is one method for locating large user profiles and evaluating them:

1. the Display Object Description (DSPOBJD) command to create an output file containing information about all the user profiles on the system:

   ```
   DSPOBJD OBJ(*ALL) OBJTYPE(*USRPRF) +
           DETAIL(*BASIC) OUTPUT(*OUTFILE)
   ```

2. Create a query program to list the name and size of each user profile, in descending sequence by size.
3. Print detailed information about the largest user profiles and evaluate the authorities and owned objects to see if they are appropriate:

   ```
   DSPUSRPRF USRPRF(user-profile-name) +
             TYPE(*OBJAUT) OUTPUT(*PRINT)
   DSPUSRPRF USRPRF(user-profile-name) +
             TYPE(*OBJOWN) OUTPUT(*PRINT)
   ```

   Some IBM-supplied user profiles are very large because of the number of objects they own. Listing and analyzing them is usually not necessary. However, you should check for programs adopting the authority of the IBM-supplied user profiles that have *ALLOBJ special authority, such as QSECOFR and QSYS.

For more information, see "IBM-Supplied User Profiles" in the iSeries Security Reference.

Audit the Security Officer's actions

A security officer or security administrator is responsible for the security on a system. A security officer has *ALLOBJ and *SECADM special authority.

You might want to keep a record of all actions performed by users with *ALLOBJ and *SECADM special authority. You can use the action auditing value in the user profile to perform this task:

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)
   ```

2. Remove the *AUDIT special authority from user profiles with *ALLOBJ and *SECADM special authority. This prevents these users from changing the auditing characteristics of their own profiles.

   Note: You cannot remove special authorities from the QSECOFR profile. Therefore, you cannot prevent a user signed on as QSECOFR from changing the auditing characteristics of that profile. However, if a user signed on as QSECOFR uses the CHGUSRAUD command to change auditing characteristics, an AD entry type is written to the audit journal.

   Recommendation: Security officers (users with *ALLOBJ or *SECADM special authority) should 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.

For more information, see "Analyzing Audit Journal Entries with Query or a Program" in the iSeries Security Reference.

Related concepts

"Special authorities" on page 20
This topic describes special authorities that can be specified for a user.

Prevent and detect security exposures

The following information is a collection of tips to help you detect potential security exposures.

Check for altered objects

This article describes how to use the Check Object Integrity (CHKOBJITG) command to look for objects that have been altered.

An altered object is usually an indication that someone is attempting to tamper with your system. You 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 has been altered and objects that have been tampered with. You can recalculate program validation values to look for objects of type *PGM, *SRVPGM, *MODULE, and *SQLPKG that have been altered. You can check the signature of objects that can be digitally signed. You can check if libraries and commands have been tampered with. You can also start a integrated file system scan or check if objects failed a previous file system scan.

You can also recalculate program validation values to look for objects of type *PGM, *SRVPGM, *MODULE, and *SQLPKG that have been altered. Running the CHKOBJITG program requires *AUDIT special authority. The command might take a long time to run because of the scans and calculations it performs. You should run it at a time when your system is not busy.

Important: To prevent impacts to either performance or system operations, distribute ownership of objects to multiple profiles. Do not assign all (or nearly all) objects to only one owner profile.

Evaluate registered exit programs

You can use the system registration function to register exit programs that should be run when certain events occur. To list the registration information on your system, type WRKREGINF OUTPUT(*PRINT).

For each exit point on the system, the report shows whether any exit programs are currently registered. When an exit point has programs that are currently registered, you can select option 8 (Display programs) from the display version of WRKREGINF to display information about the programs. Use the same method for evaluating these exit programs that you use for other exit programs and trigger programs.

Check scheduled programs

Ensure that all scheduled programs are legitimate.

The server provides several methods for scheduling jobs to run at a later time, including the job scheduler. Normally, these methods do not represent a security exposure because the user who schedules the job must have the same authority that is required to submit the job to batch. However, you should periodically check for jobs scheduled in the future. A disgruntled user who is no longer in the organization might use this method to schedule a disaster.

Check for user objects in protected libraries

Use object authority to control who can add programs to protected libraries. User objects other than programs can represent a security exposure when they are in system libraries.

Every server job has a library list. The library list determines the sequence in which the system searches for an object if a library name is not specified with the object name. For example, when you call a program without specifying where the program is, the system searches your library list in order and runs the first copy of the program that it finds.

The *iSeries Security Reference* provides more information about the security exposures of library lists and calling programs without a library name (called an unqualified call). It also provides suggestions for controlling the content of library lists and the ability to change the system library lists.

For your system to run properly, certain system libraries, such as QSYS and QGPL, must be in the library list for every job. You should use object authority to control who can add programs to these libraries. This helps to prevent someone from placing an imposter program in one of these libraries with the same name as a program that appears in a library later in the library list.

You should also evaluate who has authority to the CHGSYSLIBL command and monitor SV records in the security audit journal. A devious user could place a library ahead of QSYS in the library list and cause other users to run unauthorized commands with the same names as IBM-supplied commands.

Use the SECBATCH menu option 28 (to submit immediately) or 67 (to use the job scheduler) to run the Print User Objects (PTRUSROBJ) command. The PRTUSROBJ command prints a list of user objects (objects not created by IBM) that are in a specified library. You can then evaluate the programs on the list to determine who created them and what function they perform.

User objects other than programs can also represent a security exposure when they are in system libraries. For example, if a program writes confidential data to a file whose name is not qualified, that program might be fooled into opening an imposter version of that file in a system library.

Limit the use of adopted authority

When a program runs, the program can use adopted authority to gain access to objects in two different ways:

- The program itself can adopt the authority of its owner. This is specified in the user profile (USRPRF) parameter of the program or service program.
- The program can use (inherit) adopted authority from a previous program that is still in the job's call stack. A program can inherit the adopted authority from previous programs even if the program itself does not adopt authority. The use adopted authority (USEADPAUT) parameter of a program or a service program controls whether the program inherits adopted authority from previous programs in the program stack.

Monitor abnormal deletions

The Print Private Authorities (PRTPVTAUT) command allows you to print a report of all the private authorities for objects of a specified type in a specified library, folder, or directory.

The report lists all objects of the specified type and the users that are authorized to the object. This is a way to check for different sources of authority to objects. This command prints three reports for the selected objects. The first report (Full Report) contains all of the private authorities for each of the selected objects. The second report (Changed Report) contains additions and changes to the private authorities to the selected objects if the PRTPVTAUT command was previously run for the specified objects in the specified library, folder, or directory. Any new objects of the selected type, new authorities to existing objects, or changes to existing authorities to the existing objects are listed in the Changed Report. If the PRTPVTAUT command was not previously run for the specified objects in the specified library, folder, or directory, there will be no Changed Report. If the command has been previously run but no changes have been made to the authorities on the objects, then the Changed Report is printed but there are no objects listed.

The third report (Deleted Report) contains any deletions of privately authorized users from the specified objects since the PRTPVTAUT command was previously run. Any objects that were deleted or any users

that were removed as privately authorized users are listed in the Deleted Report. If the PRTPVTAUT command was not previously run, there will be no Deleted Report. If the command has been previously run but no delete operations have been done to the objects, then the Deleted Report is printed but there are no objects listed.

Important: You must have *ALLOBJ special authority to use this command.

Monitor abnormal system use

This article describes the task, monitor abnormal system use, explains why it is important, and provides step-by-step instructions.

The proxy server can also log all URL requests for tracking purposes. You can then review the logs to monitor use and misuse of network resources.

Monitor blatant access attempts

Monitor access to output and job queues

Sometimes a security administrator does a great job of protecting access to files and then forgets about what happens when the contents of a file are printed. Servers provide functions for you to protect sensitive output queues and job queues. You protect an output queue so that unauthorized users cannot, for example, view or copy confidential spooled files that are waiting to print. You protect job queues so that an unauthorized user cannot either redirect a confidential job to a nonconfidential output queue or cancel the job entirely.

SECBATCH menu options

24 to submit immediately 63 to use the job scheduler

The Basic system security and planning in the Information Center and Security Reference books describe how to protect your output queues and job queues. You can use the Print Queue Authority (PRTQAUT) command to print the security settings for the job queues and output queues on your system. You can then evaluate printing jobs that print confidential information and ensure that they are going to output queues and job queues that are protected.

For output queues and job queues that you consider to be security-sensitive, you can compare your security settings to the information in Appendix D of the Security Reference book: iSeries Security Reference.

Monitor for new objects installed on the system

Prevent or restrict users' from installing their own programs

When users on your system have unnecessary special authorities, your efforts to develop a good object-authority security scheme may be wasted. Object authority is meaningless when a user profile has *ALLOBJ special authority. A user with *SPLCTL special authority can see any spooled file on the system, no matter what efforts you make to secure your output queues. A user with *JOBCTL special authority can affect system operations and redirect jobs. A user with *SERVICE special authority may be able to use service tools to access data without going through the operating system.

SECBATCH menu options: 29 to submit immediately 68 to use the job scheduler

You can use the Print User Profile (PRTUSRPRF) command to print information about the special authorities and user classes for user profiles on your system. When you run the report, you have several options:

- All user profiles
- User profiles with specific special authorities

- User profiles that have specific user classes
- User profiles with a mismatch between user class and special authorities.

You can run these reports regularly to help you monitor the administration of user profiles.

Monitor for use of trigger programs

This article describes the task, monitor for use of trigger programs, explains why it is important, and provides step-by-step instructions.

DB2® UDB provides the capability to associate trigger programs with database files. Trigger-program capability is common across the industry for high-function database managers.

When you associate a trigger program with a database file, you specify when the trigger program runs. For example, you can set up the customer order file to run a trigger program whenever a new record is added to the file. When the customer's outstanding balance exceeds the credit limit, the trigger program can print a warning letter to the customer and send a message to the credit manager.

Trigger programs are a productive way both to provide application functions and to manage information. Trigger programs also provide the ability for someone with devious intentions to create a "Trojan horse" on your system. A destructive program may be sitting and waiting to run when a certain event occurs in a database file on your system.

Note: In history, the Trojan horse was a large hollow wooden horse that was filled with Greek soldiers. After the horse was introduced within the walls of Troy, the soldiers climbed out of the horse and fought the Trojans. In the computer world, a program that hides destructive functions is often called a Trojan horse.

SECBATCH menu options:

27 to submit immediately 66 to use the job scheduler

When your system ships, the ability to add a trigger program to a database file is restricted. If you are managing object authority carefully, the typical user will not have sufficient authority to add a trigger program to a database file. (Appendix D in the Security Reference book tells the authority that is required or all commands, including the Add Physical File Trigger (ADDPFTRG) command.

You can use the Print Trigger Programs (PRTTRGPGM) command to print a list of all the trigger programs in a specific library or in all libraries.

You can use the initial report as a base to evaluate any trigger programs that already exist on your system. Then, you can print the changed report regularly to see whether new trigger programs have been added to your system.

When you evaluate trigger programs, consider the following:

- Who created the trigger program? You can use the Display Object Description (DSPOBJD) command to determine this.
- What does the program do? You will have to look at the source program or talk to the program creator to determine this. For example, does the trigger program check to see who the user is? Perhaps the trigger program is waiting for a particular user (QSECOFR) in order to gain access to system resources.

After you have established a base of information, you can print the changed report regularly to monitor new trigger programs that have been added to your system.

Prevent new programs from using adopted authority

The passing of adopted authority to programs located later in the stack provides an opportunity for a knowledgeable programmer to create a Trojan horse program.

The Trojan horse program can rely on previous programs in the stack to get the authority that it needs to perform mischief. To prevent this, you can limit which users are allowed to create programs that use the adopted authority of previous programs.

When you create a new program, the system automatically sets the USEADPAUT parameter to *YES. If you do not want the program to inherit adopted authority, you must use the Change Program (CHGPGM) command or the Change Service Program (CHGSRVPGM) to set the USEADPAUT parameter to *NO.

You can use an authorization list and the use adopted authority (QUSEADPAUT) system value to control who can create programs that inherit adopted authority. When you specify an authorization list name in the QUSEADPAUT system value, the system uses this authorization list to determine how to create new programs.

When a user creates a program or service program, the system checks the user's authority to the authorization list. If the user has *USE authority, the USEADPAUT parameter for the new program is set to *YES. If the user does not have *USE authority, the USEADPAUT parameter is set to *NO. The user's authority to the authorization list cannot come from adopted authority.

The authorization list that you specify in the QUSEADPAUT system value also controls whether a user can use a CHGxxx command to set the USEADPAUT value for a program or a service program.

Note:

1. You do not need to call your authorization list QUESADPAUT. You can create an authority list with a different name. Then specify that authorization list for the QUSEADPAUT system value. In the commands in this example, substitute the name of your authorization list.
2. The QUSEADPAUT system value does not affect existing programs on your system. Use the CGHPGM command or the CHGSRVPGM command to set the USEADPAUT parameter for existing programs.

In a More Restrictive Environment: If you want most users to create new programs with the USEADPAUT parameter set to *NO, do the following:

1. 1. To set the public authority for the authorization list to *EXCLUDE, type the following: CHGAUTLE AUTL(QUSEADPAUT) USER(*PUBLIC) AUT(*EXCLUDE)
2. 2. To set up specific users to create programs that use the adopted authority of previous programs, type the following: ADDAUTLE AUTL(QUSEADPAUT) USER(user-name) AUT(*USE)

In a Less Restrictive Environment: If you want most users to create new programs with the USEADPAUT parameter set to *YES, do the following:

1. 1. Leave the public authority for the authorization list set to *USE.
2. 2. To prevent specific users from creating programs that use the adopted authority of previous programs, type the following: ADDAUTLE AUTL(QUSEADPAUT) USER(user-name) AUT(*EXCLUDE)

Use digital signatures to protect software integrity

Using digital signatures gives you greater control over which software can be loaded onto your system, and allows you more power to detect changes once it has been loaded.

All of the security precautions you take are meaningless if someone can bypass them by introducing tampered data into your system. The server has many built-in features which you can use to keep tampered software from being loaded onto your system, and to detect any such software already there. One of the techniques is **object signing**.

Object signing is the implementation of a cryptographic concept known as **digital signatures**. The idea is relatively straightforward: once a software producer is ready to ship software to customers, the producer "signs" the software. This signature does not guarantee that the software performs any specific function. However, it provides a way to prove that the software came from the producer who signed it, and that the software has not changed since it was produced and signed. This is particularly important if the software has been transmitted across the Internet or stored on media which you feel might have been modified.

The new system value, Verify Object Restore (QVFYOBJRST), provides a mechanism for setting a restrictive policy which requires all software loaded onto the system to be signed by known software sources. You can also choose a more open policy and simply verify signatures if they are present.

All i5/OS software, as well as the software for options and licensed programs, has been signed by a system trusted source. These signatures help the system protect its integrity, and they are checked when fixes are applied to the system to ensure that the fix has come from a system trusted source and that it did not change in transit. These signatures can also be checked once the software is on the system. The CHKOBJITG (Check Object Integrity) command checks signatures of the objects on the system. Additionally, the Digital Certificate Manager has panels that you can use to check signatures on objects, including objects in the operating system.

Just as the operating system has been signed, you could use digital signatures to protect the integrity of software which is critical to your business. You might buy software which has been signed by a software provider, or you might sign software which you have purchased or written. Part of your security policy, then, might be to periodically use CHKOBJITG, or the Digital Certificate Manager, to verify that the signatures on that software are still valid—that the objects have not changed since they were signed. You can also require that all software which gets restored on your system be signed by you or a known source. However, since most server software which is not produced by IBM is not currently signed so this method might be too restrictive for your system. The digital signature function gives you the flexibility to decide how best to protect your software integrity.

Modify architected transaction program names

Learn the techniques used to prevent architected transaction program names from running on the system.

Some communications requests send a specific type of signal to your system. This request is called an **architecture transaction program name** (TPN) because the name of the transaction program is part of the APPC architecture for the system. A request for display station pass-through request is an example of an architecture TPN. Architecture TPNs are a normal way for communications to function and do not necessarily represent a security exposure. However, architecture TPNs might provide an unexpected entrance into your system.

Some TPNs do not pass a profile on the request. If the request becomes associated with a communications entry whose default user is *SYS, the request may be initiated on your system. However, the *SYS profile can run system functions only, not user applications.

If you do not want architecture TPNs to run with a default profile, you can change the default user from *SYS to *NONE in communications entries.

If you do not want a specific TPN to run on your system at all, perform the following steps:

1. Create a CL program that accepts several parameters. The program should perform no function. It should simply have the Declare (DCL) statements for parameters and then end.

2. Add a routing entry for the TPN to each subsystem that has communications entries or remote location name entries. The routing entry should specify the following:
   - A Compare value (CMPVAL) value equal to the program name for the TPN with a starting position of 37.
   - A Program to call (PGM) value equal to the name of the program that you created in step 1 on page 283. This prevents the TPN from locating another routing entry, such as *ANY.

Architecture TPN requests:

This article lists the architecture transaction program names and their associated user profiles.

Table 118. Programs and users for architecture TPN requests

TPN request	Program	User profile	Description
X'30F0F8F1'	AMQCRC6A	*NONE	Message queuing
X'06F3F0F1'	QACSOTP	QUSER	APPC sign-on transaction program
X'30F0F2D1'	QANRTP	QADSM	ADSM/400 APPC configuration
X'30F0F1F9'	QCNPCSUP	*NONE	Shared folders
X'07F0F0F1'	QCNTEDDM	QUSER	DDM
X'07F6C4C2'	QCNTEDDM	QUSER	Remote SQL-DRDA1
X'30F0F7F7'	QCQNRBAS	QSVCCS	SNA CC_Server
X'30F0F1F4'	QDXPRCV	QUSER	DSNX–PC receiver
X'30F0F1F3'	QDXPSEND	QUSER	DSNX–PC sender
X'30F0F2C4'	QEVYMAIN	QUSER	ENVY**/400 Server
X'30F0F6F0'	QHQTRGT	*NONE	PC data queue
X'30F0F8F0'	QLZPSERV	*NONE	Client Access license manager
X'30F0F1F7'	QMFRCVR	*NONE	PC message receiver
X'30F0F1F8'	QMFSNDR	*NONE	PC message sender
X'30F0F6F6'	QND5MAIN	QUSER	APPN 5394 workstation controller
DB2DRDA	QCNTEDDDM	QUSER	DB2DRDA
APINGD	QNMAPINGD	QUSER	APINGD
X'30F0F5F4'	QNMEVK	QUSER	System management utilities
X'30F0F2C1'	QNPSERVR	*NONE	PWS-I network print server
X'30F0F7F9'	QOCEVOKE	*NONE	Cross-system calendar
X'30F0F6F1'	QOKCSUP	QDOC	Directory shadowing
X'20F0F0F7'	QOQSERV	QUSER	DIA Version 2
X'20F0F0F8'	QOQSERV	QUSER	DIA Version 2
X'30F0F5F1'	QOQSERV	QUSER	DIA Version 2
X'20F0F0F0'	QOSAPPC	QUSER	DIA Version 1
X'30F0F0F5'	QPAPAST2	QUSER	S/36—S/38 pass-through
X'30F0F0F9'	QPAPAST2	QUSER	Printer pass-through
X'30F0F4F6'	QPWFSTP0	*NONE	Shared folders type 2

Table 118. Programs and users for architecture TPN requests (continued)

TPN request	Program	User profile	Description
X'30F0F2C8'	QPWFSTP1	*NONE	Client access file server
X30F0F2C9''	QPWFSTP2	*NONE	Windows client access file server
X'30F0F6F9'	QRQSRVX	*NONE	Remote SQL-converged server
X'30F0F6F5'	QRQSRV0	*NONE	Remote SQL without commit
X'30F0F6F4'	QRQSRV1	*NONE	Remote SQL without commit
X'30F0F2D2'	QSVRCI	QUSER	SOC/CT
X'21F0F0F8'	QS2RCVR	QGATE	SNADS FS2 receiver
X'21F0F0F7'	QS2STSND	QGATE	SNADS FS2 sender
X'30F0F1F6'	QTFDWNLD	*NONE	PC transfer function
X'30F0F2F4'	QT1HNPCS	QUSER	TIE function
X'30F0F1F5'	QVPPRINT	*NONE	PC virtual print
X'30F0F2D3'	QWGMTP	QWGM	Ultimedia Mail/400 server
X'30F0F8F3'	QZDAINIT	QUSER	PWS-I data access server
X'21F0F0F2'	QZDRCVR	QSNADS	SNADS receiver
X'21F0F0F1'	QZDSTSND	QSNADS	SNADS sender
X'30F0F2C5'	QZHQTRG	*NONE	PWS-I data queue server
X'30F0F2C6'	QZRCSRVR	*NONE	PWS-I remote command server
X30F0F2C7''	QZSCSRVR	*NONE	PWS-I central server

Monitor access to output and job queues

This article describes how to monitor access to output and job queues, explains why it is important, and provides step-by-step instructions.

Sometimes a security administrator does a great job of protecting access to files and then forgets about what happens when the contents of a file are printed. Servers provide functions for you to protect sensitive output queues and job queues. You protect an output queue so that unauthorized users cannot, for example, view or copy confidential spooled files that are waiting to print. You protect job queues so that an unauthorized user cannot either redirect a confidential job to a nonconfidential output queue or cancel the job entirely.

You can use the following SECBATCH menu options to print the security settings for the job queues and output queues on your system: 24 to submit the job immediately and 63 to use the job scheduler. You also can use the Print Queue Authority (PRTQAUT) command to print the security settings for the job queues and output queues on your system. You can then evaluate printing jobs that print confidential information and ensure that they are going to output queues and job queues that are protected.

For more information on the PRTQAUT command, see Appendix A, and for output queues and job queues that you consider to be security-sensitive, you can compare your security settings with the required output queue and job queue function settings in the "Job Queue Commands" and "Output Queue Commands" tables in Appendix D of the iSeries Security Reference.

Monitor subsystem descriptions

This article provides suggestions for reviewing the subsystem descriptions that currently exist on your system.

When you start a subsystem on a server, the system creates an environment for work to enter the system and run. A subsystem description defines what that environment looks like. Subsystem descriptions, therefore, can provide an opportunity for devious users. A mischief-maker might use a subsystem description to start a program automatically or to make it possible to sign on without a user profile.

When you run the Revoke Public Authority (RVKPUBAUT) command, the system sets public authority to subsystem description commands to *EXCLUDE. This prevents users who are not specifically authorized (and who do not have *ALLOBJ special authority) from changing or creating subsystem descriptions.

You can use the Work with Subsystem Descriptions (WRKSBSD) command to create a list of all the subsystem descriptions. When you select **5** (Display) from the list, a menu displays for the system description that you selected. It shows a list of the parts of a subsystem environment.

You select options to see details about the parts. Use the Change Subsystem Description (CHGSBSD) command to change the first two items on the menu. To change other items, use the appropriate add, remove, or change command for the entry type. For example, to change a workstation entry, use the Change Workstation Entry (CHGWSE) command.

For additional information about working with subsystem descriptions, including lists of the shipped values for IBM-supplied subsystem descriptions, refer to the topic: Work Management.

Review autostart job entries

Look at your autostart job entries and the associated job descriptions. Ensure that you understand the function of any program that runs automatically when a subsystem starts.

An autostart job entry contains the name of a job description. The job description may contain request data (RQSDTA) that causes a program or a command to run. For example, the RQSDTA might be `CALL LIB1/PROGRAM1`. Whenever the subsystem starts, the system will run the program PROGRAM1 in library LIB1.

Review workstation names and types

Look at your workstation entries and the associated job descriptions. Ensure that no one has added or updated any entries to run programs that you are not aware of.

When a subsystem starts, it allocates all unallocated workstations that are listed (specifically or generically) in its entries for workstation names and workstations types. When a user signs on, the user is signing on to the subsystem that has allocated the workstation.

The workstation entry tells what job description will be used when a job starts at that workstation. The job description may contain request data that causes a program or a command to run. For example, the RQSDTA parameter might be `CALL LIB1/PROGRAM1`. Whenever a user signs on to a workstation in that subsystem, the system will run PROGRAM1 in LIB1.

A workstation entry might also specify a default user profile. For certain subsystem configurations, this allows someone to sign on simply by pressing the **Enter** key. If the security level (QSECURITY system value) on your system is less than `40`, you should review your workstation entries for default users.

Review job queue entries

You should periodically review the job queue entries in your subsystem descriptions to ensure that batch jobs are running where you expect them to run.

When a subsystem starts, it allocates any unallocated job queues that are listed in the subsystem description. Job queue entries do not provide any direct security exposure. However, they do provide an opportunity for someone to tamper with system performance by causing jobs to run in unintended environments.

Review routing entries

Look at the routing entries and ensure that no one has added or updated any entries to run programs that you are not aware of.

A routing entry defines what a job does once it enters the subsystem. The subsystem uses routing entries for all job types: batch, interactive, and communications jobs. A routing entry specifies the following:

- The class for the job. Like job queue entries, the class that is associated with a job can affect its performance but does not represent a security exposure.
- The program that runs when the job starts.

Review communications entries and remote location names

Ensure that the communications entries are secure.

When a communications job enters your system, the system uses the communications entries and the remote location name entries in the active subsystem to determine how the communications job will run. Look at the following for these entries:

- All subsystems are capable of running communications jobs. If a subsystem that you intend for communications is not active, a job that is trying to enter your system might find an entry in another subsystem description that meets its needs. You need to look at the entries in all subsystem descriptions.
- A communications entry contains a job description. The job description may contain request data that runs a command or program. Look at your communications entries and their associated job descriptions to ensure that you understand how jobs will start.
- A communications entry also specifies a default user profile that the system uses in some situations. Make sure that you understand the role of default profiles. If your system contains default profiles, you should ensure that they are profiles with minimal authority.

You can use the Print Subsystem Description (PRTSBSDAUT) command to identify communications entries that specify a user profile name.

For additional information about the permissions assigned to default user profiles, see: Target system assignment of user profiles for jobs.

Review prestart job entries

You should make sure that prestart job entries perform only authorized, intended functions.

You can use prestart job entries to make a subsystem ready for certain kinds of jobs so that the jobs start more quickly. Prestart jobs may start when the subsystem starts or when they are needed. Prestart job entries provide the potential for security exposures.

A prestart job entry specifies the following:

- A program to run
- A default user profile
- A job description

Review job descriptions

You should periodically review job descriptions to make sure that they do not run unintended programs. Use object authority to prevent changes to job descriptions.

Job descriptions contain request data and routing data that can cause a specific program to run when that job description is used. When the job description specifies a program in the request data parameter, the system runs the program. When the job description specifies routing data, the system runs the program that is specified in the routing entry that matches the routing data.

The system uses job descriptions for both interactive and batch jobs. For interactive jobs, the workstation entry specifies the job description. Typically, the workstation entry value is *USRPRF, so the system uses the job description that is specified in the user profile. For batch jobs, you specify the job description when you submit the job.

Job descriptions can also specify what user profile the job should run under. With security level 40 and higher, you must have *USE authority to the job description and to the user profile that is specified in the job description. With security levels lower than 40, you need *USE authority only to the job description.

You should use object authority to prevent changes to job descriptions. *USE authority is sufficient to run a job with a job description. A typical user does not need *CHANGE authority to job descriptions.

Finally, you should ensure that the default values for the Submit Job (SBMJOB) command and the Create User Profile (CRTUSRPRF) command have not been changed to point to unintended job descriptions.

Using the PRTJOBDAUT command

Use the Print Job Description Authority (PRTJOBDAUT) command to print a list of job descriptions that specify user profiles and have public authority of *USE. In the SECBATCH menu, specify either option **15** (to submit immediately) or option **54** (to use the job scheduler) to issue the PRTJOBDAUT command.

The report from the PRTJOBDAUT command shows the special authorities of the user profile that is specified in the job description. The report includes the special authorities of any group profiles that the user profile has. You can use the following command to display the user profile's private authorities:
`DSPUSRPRF USRPRF(profile-name) TYPE(*OBJAUT)`

The job description specifies the library list that the job uses when it runs. If someone can change a user's library list, that user might run an unintended version of a program in a different library. You should periodically review the library lists that are specified in the job descriptions on your system.

Monitor authority

This topic provides basic suggestions for monitoring the effectiveness of the security safeguards on your system.

A set of security reports are available to help you keep track of how the authority is set up on your system. When you run these reports initially, you can print everything (authority for all the files or for all the programs, for example).

After you have established your base of information, you can run the changed versions of reports regularly. The changed versions help you identify security-relevant changes on your system that require your attention. For example, you can run the report that shows the public authority for files every week. You can request only the changed version of the report. It will show you both new files on the system that are available to everyone and existing files whose public authority has changed since the last report.

Two menus are available to run security tools:

- Use the SECTOOLS menu for running programs interactively.
- Use the SECBATCH menu for running programs in batch. The SECBATCH menu has two parts: one for submitting jobs to the job queue immediately, and the other for placing jobs on the job scheduler.

If you are using iSeries Navigator, follow these steps to run the security tools:

1. In iSeries Navigator, expand your **Server → Security**.
2. Right-click **Policies** and select **Explore** to display a list of policies you can create and manage.

Review your security policy statement and your security memo to users as you decide which monitoring tasks you need to perform regularly. The following topics discuss several items to watch for, when monitoring authority:

Monitor authorization lists

How to use authorization lists to organize object groups based on security requirements

You can group objects with similar security requirements by using an authorization list. An authorization list contains a list of users and the authority that the users have to the objects that are secured by the list. Authorization lists provide an efficient way to manage the authority to similar objects on the system. However, in some cases, they make it difficult to keep track of authorities to objects. You can use the Print Private Authority (PRTPVTAUT) command to print information about authorities in an authorization list. The following figure shows a sample report.

```
                Private Authorities (Full Report)

SYSTEM4
Authorization  Primary                  List -------Object--------- ----------Data----------
List   Owner   Group  User     Authority Mgt Opr Mgt Exist Alter Ref Read Add Upd Dlt Execute
LIST1  QSECOFR *NONE  *PUBLIC  *EXCLUDE
LIST2  BUDNIKR *NONE  BUDNIKR  *ALL      X   X   X   X     X     X   X    X   X   X   X
                      *PUBLIC  *CHANGE   X   X   X   X     X     X
LIST3  QSECOFR *NONE  *PUBLIC  *EXCLUDE
LIST4  CJWLDR  *NONE  CJWLDR   *ALL      X   X   X   X     X     X   X    X   X   X   X
                      GROUP1   *ALL      X   X   X   X     X     X   X    X   X   X
                      *PUBLIC  *EXCLUDE
```

Figure 8. Private Authorities Report for Authorization Lists

This report shows the same information that you see on the Edit Authorization List (EDTAUTL) display. The advantage of the report is that it provides information about all authorization lists in one place. If you are setting up security for a new group of objects, for example, you can quickly scan the report to see if an existing authorization list meets your needs for those objects.

You can print a changed version of the report to see new authorization lists or authorization lists with authority changes since you last printed the report. You also have the option of printing a list of the objects that are secured by each authorization list. The following figure shows an example of the report for one authorization list:

```
                   Display Authorization List Objects

Authorization list . . . . . : CUSTAUTL
Library  . . . . . . . . . . : QSYS
Owner  . . . . . . . . . . . : AROWNER
Primary group  . . . . . . . : *NONE

                                  Primary
Object   Library   Type   Owner   Group   Text
CUSTMAS  CUSTLIB   *FILE  AROWNER *NONE
CUSTORD  CUSTORD   *FILE  OEOWNER *NONE
```

Figure 9. Report for Displaying Authorization List Objects

You can use this report, for example, to understand the effect of adding a new user to an authorization list (what authorities that user will receive).

Use authorization lists

iSeries Navigator provides security features designed to assist you in developing a security plan and policy, and configure your system to meet your company's needs. One of the functions available is the use of authorization lists.

Authorization lists have the following features.

- An authorization list group objects with similar security requirements.
- An authorization list conceptually contains a list of users and groups and the authority each has to the objects secured by the list.
- Each user and group can have a different authority to the set of object the list secures.
- Authority can be given by way of the list, rather than to individual users and groups.

Tasks that can be done using authorization lists include the following.

- Create an authorization list
- Change an authorization list.
- Add users and groups.
- Change user permissions.
- Display secured objects.

To use this function, perform the following steps:

1. From iSeries Navigator, expand your **Server → Security**. You will see **Authorization Lists and Policies**.
2. Right-click **Authorization Lists** and select **New Authorization List.** The **New Authorization List** allows you to do the following.
   - **Use:** Allows access to the object attributes and use of the object. The public may view, but not change the objects.
   - **Change:** Allows the contents of the object (with some exceptions) to be changed. v All: Allows all operations on the object, except those that are limited to the owner. The user or group can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user or group can also change ownership of the object.
   - **Exclude:** All operations on the object are prohibited. No access or operations are allowed to the object for the users and groups having this permission. Specifies the public is not allowed to use the object.

When working with authorization lists you will want to grant permissions for both objects and data. Object permissions you can choose are listed below:

- **Operational:** Provides the permission to look at the description of an object and use the object as determined by the data permission that the user or group has to the object.
- **Management:** Provides the permission to specify the security for the object, move or rename the object, and add members to the database files.
- **Existence:** Provides the permission to control the object's existence and ownership. The user or group can delete the object, free storage of the object, perform save and restore operations for the object, and transfer ownership of the object. If a user or group has special save permission, the user or group does not need object existence permission.
- **Alter:** For database files and SQL packages only, provides the permission needed to alter the attributes of an object. If the user or group has this permission on a database file, the user or group can add and remove triggers, add and remove referential and unique constraints, and change the attributes of the database file. If the user or group has this permission on an SQL package, the user or group can change the attributes of the SQL package. This permission is currently used only for database files and SQL packages.
- **Reference:** For database files and SQL packages only, provides the permission needed to reference an object from another object such that operations on that object may be restricted by the other object. If

the user or group has this permission on a physical file, the user or group can add referential constraints in which the physical file is the parent. This permission is currently used only for database files.

Data permissions you can choose are listed below.

- **Read:** Provides the permission needed to get and display the contents of the object, such as viewing records in a file.
- **Add:** Provides the permission to add entries to an object, such as adding messages to a message queue or adding records to a file.
- **Update:** Provides the permission to change the entries in an object, such as changing records in a file. v Delete: Provides the permission to remove entries from an object, such as removing messages from a message queue or deleting records from a file.
- **Execute:** Provides the permission needed to run a program, service program or SQL package. The user can also locate an object in a library or directory.

For information on monitoring authority on the server, see: Monitor private authority to objects.

Monitor private authority to objects

This article describes the SECBATCH menu options and security commands that you can use to monitor private authority to objects.

Private authority is the authority specifically given to a user for an object that overrides any other authorities, such as the authority of a user's group profile or an authorization list. Users who are not listed in the group profile or authorization list cannot access objects with private authority.

You can use the following SECBATCH menu options to monitor private authority to objects: 12 to submit immediately and 14 to use the job scheduler. SECBATCH menu contains options for the object types that are typically of concern to security administrators. Use the general options (19 and 58) to specify the object type.

In addition, you can use the Print Private Authority (PRTPVTAUT) command to print a list of all the private authorities for objects of a specified type in a specified library. This report can help you detect new authorities to objects. It can also help you keep your private authority scheme from becoming convoluted and unmanageable.

For information on monitoring authority on the server, see: Monitor public authority to objects.

Monitor public authority to objects

This article describes the SECBATCH menu options and security commands that you can use to monitor public authority to objects.

Public authority is the authority for an object granted to all users.

For both simplicity and performance, most systems are set up so that most objects are available to most users. Users are explicitly denied access to certain confidential, security-sensitive objects rather than having to be explicitly authorized to use every object. A few systems with high security requirements take the opposite approach and authorize objects on a need-to-know basis. On those systems, most objects are created with the public authority set to *EXCLUDE.

This is an object-based system with many different types of objects. Most object types do not contain sensitive information or perform security-relevant functions. As a security administrator on a system with typical security needs, you probably want to focus your attention on objects that require protection, such as database files and programs. For other object types, you can just set public authority that is sufficient for your applications, which for most object types is *USE authority.

You can use the Print Public Authority (PRTPUBAUT) command to print information about objects that public users can access. (A public user is anyone with signon authority who does not have explicit authority to an object.) When you use the PRTPUBAUT command, you can specify the object types, and libraries or directories, that you want to examine.

You can use options 11 or 50 on the SECBATCH menu to print the Publicly Authorized Objects report for the object types that might have security implications. Use the general options (18 and 57) to specify the object type. You can print the changed version of this report regularly to see what objects might require your attention.

For more information, see: Monitor special authorities.

Monitor user environments

This article discusses using the SECBATCH menu and commands to monitor user environments.

One role of the user profile is to define the environment for the user, including the output queue, the initial menu, and the job description. The user's environment affects how the user sees the system and, to some extent, what the user is allowed to do. The user must have authority to the objects that are specified in the user profile. However, if your authority scheme is still in progress or is not very restrictive, the user environment that is defined in a user profile may produce results that you do not intend.

Use the following SECBATCH menu options to monitor user environments: 29 to submit the job immediately and 68 to use the job scheduler:

- The user's job description may specify a user profile that has more authority than the user.
- The user may have an initial menu that does not have a command line. However, the user's attention-key-handling program may provide a command line.
- The user may be authorized to run confidential reports. However, the user's output may be directed to an output queue that is available to users who should not see the reports.

You can use the *ENVINFO option of the Print User Profile (PRTUSRPRF) command to help you monitor the environments that are defined for system users. The following figure shows an example of the report:

```
                    User Profile Information

Report type . . . . . . . . . . : *ENVINFO
Select by . . . . . . . . . . . : *USRCLS
                  Initial Initial   Job           Message  Output   Attention
User      Current Menu/   Program/  Description/  Queue/   Queue/   Program/
Profile   Library Library Library   Library       Library  Library  Library
AUDSECOFR AUDITOR MAIN    *NONE     QDFTJOBD      QSYSOPR  *WRKSTN  *SYSVAL
                  *LIBL             QGPL          QSYS
USERA     *CRTDFT OEMENU  *NONE     QDFTJOBD      USERA    *WRKSTN  *SYSVAL
                  *LIBL             QGPL          QUSRSYS
USERB     *CRTDFT INVMENU *NONE     QDFTJOBD      USERB    *WRKSTN  *SYSVAL
                  *LIBL             QGPL          QUSRSYS
USERC     *CRTDFT PAYROLL *NONE     QDFTJOBD      USERC    PAYROLL  *SYSVAL
                  *LIBL             QGPL          QUSRSYS  PRPGMLIB
```

Figure 10. Print User Profile: User Environment Report

For more information, see: Monitor security messages.

Monitor special authorities

This topic describes the SECBATCH menu options and commands used to monitor special authorities.

Special authority is a type of authority a user can have to perform system functions, including all object authority, save system authority, job control authority, security administrator authority, spool control authority, service authority, and system configuration authority.

When users on your system have unnecessary special authorities, your efforts to develop a good object-authority scheme may be wasted. Object authority is meaningless when a user profile has *ALLOBJ special authority. A user with *SPLCTL special authority can see any spooled file on the system, no matter what efforts you make to secure your output queues. A user with *JOBCTL special authority can affect system operations and redirect jobs. A user with *SERVICE special authority may be able to use service tools to access data without going through the operating system.

Use the following SECBATCH menu options to monitor special authorities: 29 to submit the job immediately or 68 to use the job scheduler.

You can use the Print User Profile (PRTUSRPRF) command to print information about the special authorities and user classes for user profiles on your system. When you run the report, you have several options:

- All user profiles
- User profiles with specific special authorities
- User profiles that have specific user classes
- User profiles with a mismatch between user class and special authorities.

The following figure shows an example of the report that shows the special authorities for all user profiles:

```
                     User Profile Information

Report type . . . . . . . . . . : *AUTINFO
Select by . . . . . . . . . . . : *SPCAUT
Special authorities . . . . . : *ALL
              -------------Special Authorities-------------
                           *IO                                          Group
User    Group    *ALL *AUD SYS *JOB *SAV *SEC *SER *SPL User            Group     Authority Limited
Profile Profiles OBJ  IT   CFG CTL  SYS  ADM  VICE  CTL Class   Owner   Authority Type      Capability
USERA  *NONE     X    X    X   X    X    X    X     X   *SECOFR *USRPRF *NONE     *PRIVATE  *NO
USERB  *NONE                    X    X                    *PGMR   *USRPRF *NONE     *PRIVATE  *NO
USERC  *NONE     X    X    X   X    X    X    X     X   *SECOFR *USRPRF *NONE     *PRIVATE  *NO
USERD  *NONE                                              *USER   *USRPRF *NONE     *PRIVATE  *NO
```

Figure 11. User Information Report: Example 1

In addition to the special authorities, the report shows the following:

- Whether the user profile has limited capability.
- Whether the user or the user's group owns new objects that the user creates.
- What authority the user's group automatically receives to new objects that the user creates.

The following figure shows an example of the report for mismatched special authorities and user classes. Notice the following:

- USERX has a system operator (*SYSOPR) user class but has *ALLOBJ and *SPLCTL special authorities.
- USERY has a user (*USER) user class but has *SECADM special authority.
- USERZ also has a user (*USER) class and *SECADM special authority. You can also see that USERZ is a member of the QPGMR group, which has *JOBCTL and *SAVSYS special authorities.

```
                    User Profile Information

Report type . . . . . . . . . . : *AUTINFO
Select by . . . . . . . . . . . : *MISMATCH
             -------------Special Authorities-------------
                          *IO                                                Group
User    Group    *ALL *AUD SYS *JOB *SAV *SEC *SER *SPL User            Group     Authority Limited
Profile Profiles OBJ  IT   CFG CTL  SYS  ADM  VICE  CTL Class   Owner   Authority Type      Capability
USERX   *NONE    X             X    X               X   *SYSOPR *USRPRF *NONE     *PRIVATE  *NO
USERY   *NONE                              X             *USER   *USRPRF *NONE     *PRIVATE  *NO
USERZ   *NONE                              X             *USER   *USRPRF *NONE     *PRIVATE  *NO
        QPGMR                 X    X
```

Figure 12. User Information Report: Example 2

You can run these reports regularly to help you monitor the administration of user profiles.

For more information, see: Monitor user environments.

Related concepts

"Special authorities" on page 20
This topic describes special authorities that can be specified for a user.

Monitor sign-on and password activity

If you are concerned about unauthorized attempts to enter your system, you can use the PRTUSRPRF command to help you monitor sign-on and password activity.

Here are several suggestions for using this report:

- Determine whether the password expiration interval for some user profiles is longer than the system value and whether the longer expiration interval is justified. For example, in the report, USERY has a password expiration interval of 120 days.
- Run this report regularly to monitor unsuccessful signon attempts. Someone who is trying to break into your system may be aware that your system takes action after a certain number of unsuccessful attempts. Each night, the would-be intruder might try fewer times than your QMAXSIGN value to avoid alerting you to the attempts. However, if you run this report early each morning and notice that certain profiles often have unsuccessful signon attempts, you might suspect that you have a problem.
- Identify user profiles that have not been used for a long time or whose passwords have not been changed for a long time.

Monitor user profile activity

As a security administrator, you need to control and audit changes that occur to user profiles on your system.

User profiles provide entry to your system. Parameters in the user profile determine a user's environment and a user's security characteristics.

You can set up security auditing so that your system writes a record of changes to user profiles. You can use the DSPAUDJRNE command to print a report of those changes. You can create exit programs to evaluate requested actions to user profiles.

The following table shows the exit points that are available for user profile commands.

Table 119. Exit points for user profile activity

User profile command	Exit point name
Create User Profile (CRTUSRPRF)	QIBM_QSY_CRT_PROFILE
Change User Profile (CHGUSRPRF)	QIBM_QSY_CHG_PROFILE
Delete User Profile (DLTUSRPRF)	QIBM_QSY_DLT_PROFILE

Table 119. Exit points for user profile activity (continued)

User profile command	Exit point name
Restore User Profile (RSTUSRPRF)	QIBM_QSY_RST_PROFILE

Your exit program can, for example, look for changes that might cause the user to run an unauthorized version of a program. These changes might be assigning either a different job description or a new current library. Your exit program might either notify a message queue or take some action (like changing or disabling the user profile) based on the information that the exit program receives.

The *Security Reference* book provides more information about the exit programs for user profile actions. See: iSeries Security Reference.

Monitor security messages

How to monitor security messages and why they are important

Some security-relevant events, such as incorrect signon 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.

Prevent loss of auditing information

This article describes which information to look for to prevent loss of auditing information.

Two system values control what the system does when error conditions may cause the loss of audit journal entries.

Audit Force Level: The QAUDFRCLVL system value determines how often the system writes audit journal entries from memory to auxiliary storage. The QAUDFRCLVL system value works like the force level for database files. You should follow similar guidelines in determining the correct force level for your installation.

If you allow the system to determine when to write entries to auxiliary storage, it balances the performance impact against the potential loss of information in a power outage. *SYS is the default and the recommended choice.

If you set the force level to a low number, you minimize the possibility of losing audit records, but you may notice a negative performance impact. If your installation requires that no audit records be lost in a power failure, you must set the QAUDFRCLVL to 1.

Audit End Action: The QAUDENDACN system value determines what the system does if it is unable to write an entry to the audit journal. The default value is *NOTIFY. The system does the following if it is unable to write audit journal entries and QAUDENDACN is *NOTIFY:

1. The QAUDCTL system value is set to *NONE to prevent additional attempts to write entries.
2. Message CPI2283 is sent to the QSYSOPR message queue and the QSYSMSG message queue (if it exists) every hour until auditing is successfully restarted.
3. Normal processing continues.
4. If an IPL is performed on the system, message CPI2284 is sent to the QSYSOPR and QSYSMSG message queues during the IPL.

Note: In most cases, performing an IPL resolves the problem that caused auditing to fail. After you have restarted your system, set the QAUDCTL system value to the correct value. The system attempts to write an audit journal record whenever this system value is changed.

You can set the QAUDENDACN to power down your system if auditing fails (*PWRDWNSYS). Use this value only if your installation requires that auditing be active for the system to run. If the system is unable to write an audit journal entry and the QAUDENDACN system value is *PWRDWNSYS, the following happens:

1. The system powers down immediately (the equivalent of issuing the PWRDWNSYS *IMMED command).
2. SRC code `B900 3D10` is displayed.

Next, you must do the following:

1. Start an IPL from the system unit. Make sure that the device specified in the system console (QCONSOLE) system value is powered on.
2. To complete the IPL, a user with *ALLOBJ and *AUDIT special authority must sign on at the console.
3. The system starts in a restricted state with a message indicating that an auditing error caused the system to stop.
4. The QAUDCTL system value is set to *NONE.
5. To restore the system to normal, set the QAUDCTL system value to a value other than `NONE`.

   When you change the QAUDCTL system value, the system attempts to write an audit journal entry. If it is successful, the system returns to a normal state. If the system does not successfully return to a normal state, use the job log to determine why auditing has failed. Correct the problem and attempt to reset the QAUDCTL value again.

Manage the journal receivers

How to manage the journal receivers.

If you choose to manage journal receivers manually, use the following procedure to detach, save and delete a journal receiver:

1. Type `CHGJRN JRN(QAUDJRN) JRNRCV(*GEN)`. This command does the following:
   a. Detaches the currently attached receiver.
   b. Creates a new receiver with the next sequential number.
   c. Attaches the new receiver to the journal.

      For example, if the current receiver is AUDRCV0003, the system creates and attaches a new receiver called AUDRCV0004.

      The Work with Journal Attributes (WRKJRNA) command tells you which receiver is currently attached: `WRKJRNA QAUDJRN`.
2. Use the Save Object (SAVOBJ) command to save the detached journal receiver. Specify object type *JRNRCV.
3. Use the Delete Journal Receiver (DLTJRNRCV) command to delete the receiver. If you try to delete the receiver without saving it, you receive a warning message.

Use audit journals to monitor object activity

You can use the audit journal to monitor object activity and to log security events.

When you want to analyze the audit information you have collected in the QAUDJRN journal, you can use the Display Journal (DSPJRN) command. With this command, information from the QAUDJRN journal can be written to a database file. An application program or a query tool can be used to analyze the data.

If you include the *AUTFAIL value for system action auditing (the QAUDLVL system value), the system writes an audit journal entry for every unsuccessful attempt to access a resource. For critical objects, you can also set up object auditing so the system writes an audit journal entry for each successful access.

The audit journal records only that the object was accessed. It does not log every transaction to the object. For critical objects on your system, you may want more detailed information about the specific data that was accessed and changed. Object journaling can provide you with those details. Object journaling is used primarily for object integrity and recovery. A security officer or auditor can also use these journal entries to review object changes. *Do not* journal any objects to the QAUDJRN journal.

Journal entries can include:

- Identification of the job and user and the time of access
- Before- and afterimages of all object changes
- Records of when the object was opened, closed, changed, and saved

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 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.

Managing the audit journal and journal receivers

The auditing journal, QSYS/QAUDJRN, is intended *solely* for security auditing. Objects should not be journaled to the audit journal. Commitment control should not use the audit journal. User entries should not be sent to this journal using the Send Journal Entry (SNDJRNE) command or the Send Journal Entry (QJOSJRNE) API.

Special locking protection is used to ensure that the system can write audit entries to the audit journal. When auditing is active (the QAUDCTL system value is not *NONE), the system arbitrator job (QSYSARB) holds a lock on the QSYS/QAUDJRN journal. You cannot perform certain operations on the audit journal when auditing is active, such as:

- DLTJRN command
- ENDJRN*xxx* command
- APYJRNCHG command
- RMVJRNCHG command
- DMPOBJ or DMPSYSOBJ command
- Moving the journal
- Restoring the journal
- Operations that work with authority, such as the GRTOBJAUT command
- WRKJRN command

All security entries in the audit journal have a journal code of T. In addition to security entries, system entries also appear in the journal QAUDJRN. These are entries with a journal code of J, which relate to initial program load (IPL) and general operations performed on journal receivers (for example, saving the receiver).

If damage occurs to the journal or to its current receiver so that the auditing entries cannot be journaled, the QAUDENDACN system value determines what action the system takes. Recovery from a damaged journal or journal receiver is the same as for other journals.

You may want to have the system manage the changing of journal receivers. Specify MNGRCV(*SYSTEM) when you create the QAUDJRN journal, or change the journal to that value. If you

specify MNGRCV(*SYSTEM), the system automatically detaches the receiver when it reaches its threshold size and creates and attaches a new journal receiver. This is called *system change-journal management*.

If you specify MNGRCV(*USER) for the QAUDJRN, a message is sent to the threshold message queue specified for the journal when the journal receiver reaches a storage threshold. The message indicates that the receiver has reached its threshold. Use the CHGJRN command to detach the receiver and attach a new journal receiver. This prevents Entry not journaled error conditions. If you do receive a message, you must use the CHGJRN command for security auditing to continue.

The default message queue for a journal is QSYSOPR. If your installation has a large volume of messages in the QSYSOPR message queue, you may want to associate a different message queue, such as AUDMSG, with the QAUDJRN journal. You can use a message handling program to monitor the AUDMSG message queue. When a journal threshold warning is received (CPF7099), you can automatically attach a new receiver. If you use system change-journal management, then message CPF7020 is sent to the journal message queue when a system change journal is completed. You can monitor for this message to know when to do a save of the detached journal receivers.

Attention: The automatic cleanup function provided using Operational Assistant menus does not clean up the QAUDJRN receivers. You should regularly detach, save, and delete QAUDJRN receivers to avoid problems with disk space. See the Journal management topic for complete information about managing journals and journal receivers.

Note: The QAUDJRN journal is created during an IPL if it does not exist and the QAUDCTL system value is set to a value other than *NONE. This occurs only after an unusual situation, such as replacing a disk device or clearing an auxiliary storage pool.

For more information on the audit journal entries, see “Appendix F” in the iSeries Security Reference.

Save and delete audit journal receivers

This article describes how to save and delete audit journal receivers, explains why it is important, and provides step-by-step instructions.

Purpose:

To attach a new audit journal receiver; to save and delete the old receiver

How to:

- CHGJRN QSYS/QAUDJRN
- JRNRCV(*GEN) SAVOBJ (to save old receiver)
- DLTJRNRCV (to delete old receiver)

Authority:

*ALL authority to journal receiver; *USE authority to journal

Note: Select a time when the system is not busy to save and delete audit journal receivers.

You should regularly detach the current audit journal receiver and attach a new one for two reasons:

- Analyzing journal entries is easier if each journal receiver contains the entries for a specific, manageable time period.
- Large journal receivers can affect system performance, in addition to taking valuable space on auxiliary storage.

Having the system manage receivers automatically is the recommended approach. You can specify this by using the *Manage receiver* parameter when you create the journal.

If you have set up action auditing and object auditing to log many different events, you may need to specify a large threshold value for the journal receiver. If you are managing receivers manually, you may

need to change journal receivers daily. If you log only a few events, you may want to change receivers to correspond with the backup schedule for the library containing the journal receiver.

You use the CHGJRN command to detach a receiver and attach a new receiver.

System-Managed Journal Receivers: If you have the system manage the receivers, use the following procedure to save all detached QAUDJRN receivers and to delete them:

1. Type `WRKJRNA QAUDJRN`. The display shows you the currently attached receiver. Do not save or delete this receiver.
2. Use F15 to work with the receiver directory. This shows all receivers that have been associated with the journal and their status.
3. Use the SAVOBJ command to save each receiver, except the currently attached receiver, which has not already been saved.
4. Use the DLTJRNRCV command to delete each receiver after it is saved.

Note: An alternative to the above procedure could be done using the journal message queue and monitoring for the CPF7020 message which indicates that the system change journal has completed successfully.

For additional information, refer to the section: Backup and recovery.

Stop the audit function

How to turn off the auditing function

You may want to use the audit function periodically rather than all the time. For example, you might want to use it when testing a new application. Or you might use it to perform a quarterly security audit. To stop the auditing function, do the following:

1. Use the WRKSYSVAL command to change the QAUDCTL system value to *NONE. This stops the system from logging any more security events.
2. Detach the current journal receiver using the CHGJRN command.
3. Save and delete the detached receiver, using the SAVOBJ and DLTJRNRCV commands.
4. You can delete the QAUDJRN journal once you change QAUDCTL to *NONE. If you plan to resume security auditing in the future, you may want to leave the QAUDJRN journal on the system.

However, if the QAUDJRN journal is set up with MNGRCV(*SYSTEM), the system detaches the receiver and attaches a new one whenever you perform an IPL, whether or not security auditing is active. You need to delete these journal receivers. Saving them before deleting them should not be necessary because they do not contain any audit entries.

Use the history log

How to use the history log, why it is important, and step-by-step instructions for setting it up.

Some security-related events, such as exceeding the incorrect signon 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.

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.

Related information for security planning

Listed here are the product manuals and IBM Redbooks™ (in PDF format), Web sites, and information center topics that relate to the Plan and set up system security topic. You can view or print any of the PDFs.

Manuals

iSeries Security Reference  (13 682 KB)

Other information

- Intrusion detection describes how to prevent intrusions that come in over the TCP/IP network.

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)  .

Appendix. 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:

| IBM World Trade Asia Corporation
| Licensing
| 2-31 Roppongi 3-chome, Minato-ku
| Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country 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 information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, IBM License Agreement for Machine Code, 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.

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.

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. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

- AIX
- AS/400
- DRDA

| eServer
| i5/OS
| IBM
| iSeries
| Java
| Linux
| Lotus Notes
| Microsoft
| Net.Data
| NetServer)
| OS/400
| PowerPC
| Redbooks
| System/36
| System/38
| UNIX
| Windows
| xSeries
| z/OS

Microsoft, Windows, Windows NT®, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

| Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

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.

Printed in USA